



Pemodelan Matematis,
Departemen Matematika UI Semester Genap TA
2021/2022

Stock Trading Robot

Kelompok B3

Anatasya Oktaviani Handriati | Ashley Kainama |
Christantina Ethan Agustya | Halwatunnisa |
Kezya Samantha Sherryn | Marsha Putri Mahira | M. Shiqo Filla |
Muhammad Ichsanudin | Muhammad Reza Maullanna |
Prasetya Nugroho Hutomo | Richardy Lobo' Sapan

Latar Belakang Masalah



Banyaknya kasus mengenai kerugian akibat trading saham membuat para trader mulai ragu dalam pengelolaan uang di bidang saham. Selain itu, keuntungan yang diperoleh tidaklah terbaca jelas dalam hal trading saham bagi pemula. Maka dari itu, diperlukan suatu sarana yang dapat membantu dalam menganalisis data mengenai saham sehingga meminimalisir terjadinya *error* pada proses perdagangan saham di Indonesia.

Masalah Trader dalam *Stock Trading* (*definisi masalah*)

1. Kesulitan memilih pembelian saham yang terbaik, khususnya pembelian saham di Indonesia
2. Minimnya analisis mengenai saham yang harus dibeli dan dijual
3. Belum memahami resiko/laju jalannya saham

Input dan Output Stock Trading Robot



Input:

- Saham di suatu periode
- Budget trader

Output:

- Trend dari saham
- Rekomendasi pembelian saham (khususnya di indonesia)

Note: Saham dibatasi pada saham yg ada di Bursa Efek Indonesia

Apa itu Saham (Stock)?



“Sebuah **surat berharga** yang berisi **bukti kepemilikan** ataupun penyertaan dari seorang atau instansi perusahaan”

--Sapto Raharjo

“Surat **bukti atau tanda bukti kepemilikan individu atau instansi** pada suatu perusahaan perseroan terbatas.”

--Tjiptono Darmadji dan Hendy M. Fakhrudin

“Saham adalah **suatu alat bukti atas kepemilikan** dari **sebuah perusahaan** yang berupa lembaran-lembaran surat berharga”

Jenis-jenis Saham



Berdasarkan sektornya;

- energi,
- barang baku,
- perindustrian,
- barang konsumen primer,
- barang konsumen non-primer,
- kesehatan,
- keuangan,
- properti & real estate,
- teknologi,
- infrastruktur,
- perdagangan jasa dan investasi,
- transportasi & logistik, dan
- produk investasi tercatat.

Jenis-jenis Saham



Berdasarkan kemampuannya

1. Saham Biasa (*Common Stock*)

Ciri-ciri:

- ❖ Setiap pemegang saham memiliki hak suara yang sama dalam hal memilih dewan komisaris
- ❖ Setiap hak pemegang saham akan diprioritaskan saat perusahaan akan mengeluarkan saham baru
- ❖ Setiap pemegang saham punya tanggung jawab yang sifatnya terbatas (sebesar nilai saham yang sudah disetorkannya)

Jenis-jenis Saham



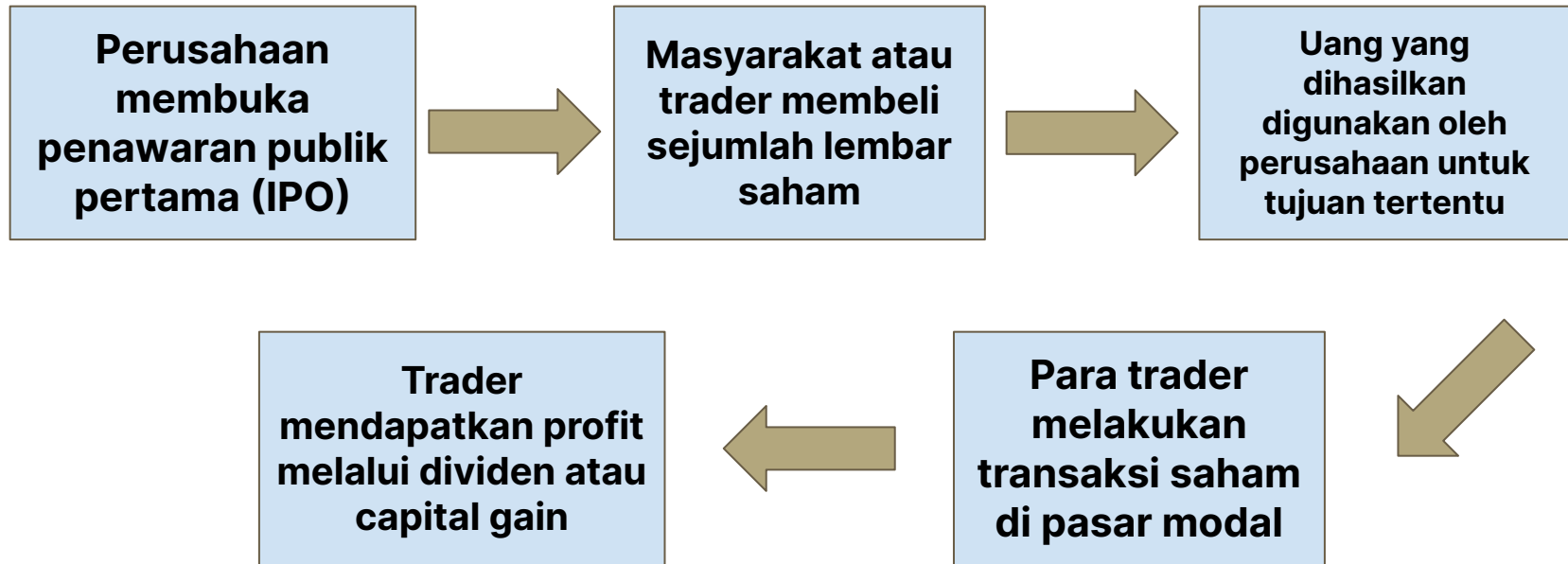
Berdasarkan kemampuannya

2. Saham Preferen (*Preferred Stock*)

Ciri-ciri:

- ❖ Memiliki beberapa tingkatan dengan karakteristik yang berbeda
- ❖ Ada tagihan atas suatu pendapatan, memiliki prioritas tinggi dalam hal pembagian nilai dividen
- ❖ Bisa ditukarkan dengan saham biasa, dengan kesepakatan yang ada

Alur Transaksi Saham



Analisis Masalah: Cara Memilih Saham

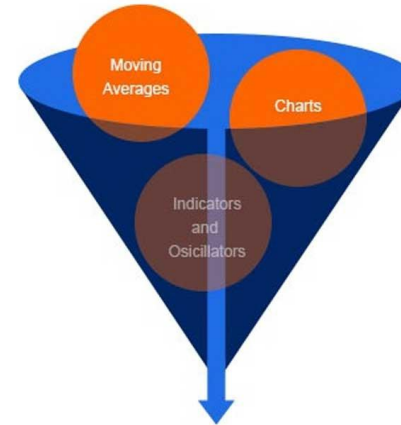


Fundamental Analysis



VS

Technical Analysis



Analisis Masalah: Cara Memilih Saham



Fundamental Analysis

Analisa yang bertujuan mempelajari **hal-hal yang berhubungan dengan kondisi keuangan perusahaan** tertentu.

Technical Analysis

Analisa yang **berdasarkan pada data harga historis yang terjadi pada pasar saham**. Sesuai namanya, analisa teknikal melibatkan pendekatan teknis dengan penggunaan dua metode. Metode klasik (chart harga saham), sedangkan metode modern (indikator dengan algoritma atau rumus statistik).

Apa itu Stock Trading Robot?



Sistem perdagangan otomatis (trading robot) merupakan sistem memungkinkan pedagang untuk menetapkan aturan khusus untuk entri dan keluar perdagangan yang, setelah diprogram, dapat dieksekusi secara otomatis melalui komputer.

- Dana kuantitatif yang dikelola melalui strategi perdagangan sistematis terkomputerisasi, sering disebut sebagai **robot atau bot investasi**, adalah kategori dana yang tumbuh paling cepat.
- Faktanya, berbagai platform melaporkan **70%-80% atau lebih saham** yang diperdagangkan di bursa saham AS berasal dari sistem perdagangan otomatis.

Analisis Masalah: Peran Stock Trading Robot



- Mengambil dan menganalisa data saham
- Membuat *Sell/Buy Order*
- Menentukan waktu yang tepat untuk *Buy/Sell* sesuai dengan algoritma tradingnya



Cara Kerja Stock Trading Robot

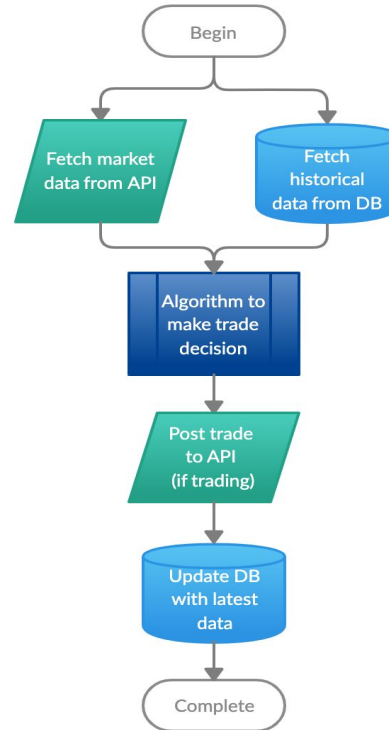


Robot akan mencari **peluang *open trade*, *open sell*, dan *buy*** di pasar saham. Lalu itu akan **menjalankan algoritma** berdasarkan serangkaian sinyal pasar saham yang membantu kita dalam **menentukan apakah akan membeli atau menjual** saham pada titik waktu tertentu.

Cara Kerja Stock Trading Robot



Steps for market XYZ



Kesimpulan



- Saham adalah **suatu alat bukti atas kepemilikan** dari **sebuah perusahaan** yang berupa lembaran-lembaran surat berharga
- Robot akan mencari **peluang *open trade*, *open sell*, dan *buy*** di pasar saham. Lalu membantu kita dalam **menentukan apakah akan membeli atau menjual** saham pada titik waktu tertentu.
- Metode yang dijelaskan hanya mencakup *technical analysis*. Untuk melakukan jual beli saham, **sangat dianjurkan** untuk melihat juga dari fundamental analysis dan menggunakan *trading strategy*, yang mana itu di luar cakupan proyek ini.

Referensi



Best 10 Crypto Trading Bots For 2022 [Review And Comparison]. (2022, March 7). Software Testing Help. <https://www.softwaretestinghelp.com/best-crypto-trading-bots/#:~:text=They%20use%20automated%20and%20smart,bots%2C%20and%20market%20maker%20bots.>

Build a Crypto Trading Bot | Tips and Best Practices. (2021, January 12). DevTeam.Space; DTS. <https://www.devteam.space/blog/how-to-build-a-crypto-trading-bot/>

Building Profitable Algorithmic Trading Bots. (2021, June 29). Trality Blog. <https://www.trality.com/blog/building-algorithmic-trading-bots>

Coding Your Own Algo-Trading Robot. (2022). Investopedia. <https://www.investopedia.com/articles/active-trading/081315/how-code-your-own-algo-trading-robot.asp#toc-backtesting-and-optimization>

LSTM for Text Classification | Beginners Guide to Text Classification. (2021, June 14). Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/#:~:text=it%20be%20used%3F-,What%20is%20LSTM%3F,patterns%20LSTMs%20perform%20fairly%20better>

Predicting Stock Prices Using Machine Learning - neptune.ai. (2021, July 29). Neptune.ai. <https://neptune.ai/blog/predicting-stock-prices-using-machine-learning>

Schmitz, J. (2020, May 17). *The beginning of a deep learning trading bot — Part1: 95% accuracy is not enough.* Medium; Towards Data Science. <https://towardsdatascience.com/the-beginning-of-a-deep-learning-trading-bot-part1-95-accuracy-is-not-enough-c338abc98fc2>

Folger, Jean. Automated Trading Systems: The Pros and Cons
. <https://www.investopedia.com/articles/trading/11/automated-trading-systems.asp>. Diakses pada 14-3-2022

Kowlakowski, Mark. How Robots Rule the Stock Market (SPX, DJIA)

.
[https://www.investopedia.com/news/how-robots-rule-stock-market-spx-djia/#:~:text=Robots%20apparently%20rule%20the%20stock,\(CS\)%20reported%20by%20Bloomberg.](https://www.investopedia.com/news/how-robots-rule-stock-market-spx-djia/#:~:text=Robots%20apparently%20rule%20the%20stock,(CS)%20reported%20by%20Bloomberg.) Diakses pada 14-3-2022



Thank You !!





Appendix



Metode yang Digunakan Oleh Stock Bot



Ada 2 metode yang digunakan dalam pemodelan ini:

1. Moving Average (MA)

MA adalah **rata-rata yang bergerak sepanjang skala waktu/periode**; titik data yang lebih lama akan dihapus saat titik data yang lebih baru ditambahkan.

Periode yang umum digunakan adalah MA **20 hari, 50 hari, dan 200 hari** untuk investasi jangka pendek, jangka menengah, dan jangka panjang.

2. Long Short-term Memory (LSTM)

Singkatnya, komponen kunci untuk memahami model LSTM adalah **Cell State (Ct)**, yang mewakili ingatan short term dan long term internal sel.

Metode yang Digunakan Oleh Stock Bot



1. Moving Average (MA)

- Dua jenis MA yang paling disukai oleh analis keuangan: **Simple MA** dan **Exponential MA**.
- **SMA**, kependekan dari **Simple Moving Average**, menghitung rata-rata kisaran harga saham (penutupan) selama sejumlah periode tertentu dalam kisaran tersebut.

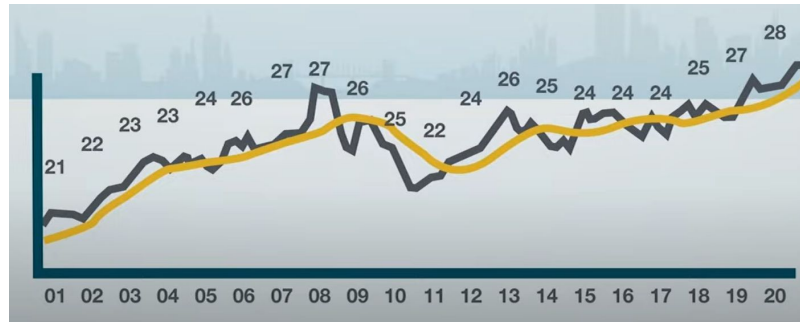
$$SMA = \frac{P1 + P2 + ... + Pn}{N}$$

- P_n = Harga Saham pada waktu $t = n$
- N = Jumlah titik waktu

Metode yang Digunakan Oleh Stock Bot



1. Moving Average (MA)



- Sebagai contoh, ambil periode 20 hari yang pertama, maka kita dapat persamaan:

$$\frac{21 + 22 + 23 + 23 + 24 + 26 + 27 + 27 + 26 + 25 + 22 + 24 + 26 + 25 + 24 + 24 + 24 + 25 + 27 + 28}{20}$$

- Jika ada data baru, yaitu data hari ke-21, maka data hari terlama, yaitu hari ke-1 akan dibuang dan hari-21 ditambahkan

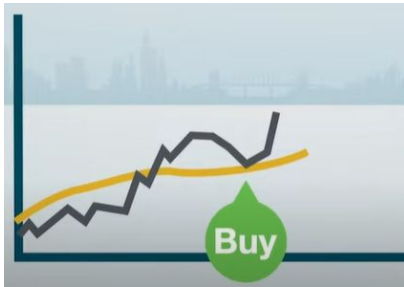
$$\frac{\cancel{21} + 22 + 23 + 23 + 24 + 26 + 27 + 27 + 26 + 25 + 22 + 24 + 26 + 25 + 24 + 24 + 24 + 25 + 27 + 28 + 29}{20}$$

Metode yang Digunakan Oleh Stock Bot

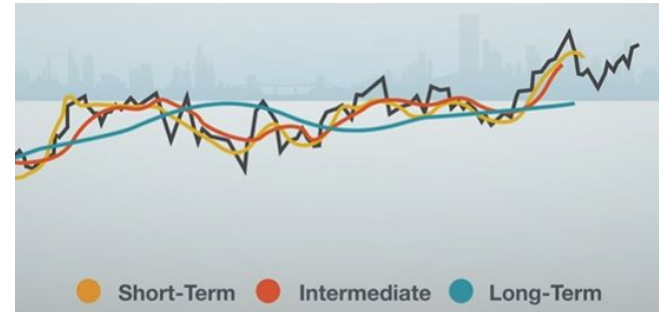


1. Moving Average (MA)

- Selain untuk mengetahui trend dari saham, MA jg digunakan untuk mengetahui waktu jual & beli



- Saat ada kurva yang menurun lalu menaik, tanda untuk membeli
- Saat ada kurva yang menaik lalu menurun, tanda untuk menjual



- Semakin lama periode yang dipilih, maka semakin 'smooth' grafik yang diperoleh -> Semakin jarang aktivitas jual & beli saham
- Kelemahan: Sering terjadi 'lag' antara SMA dan real-life

Metode yang Digunakan Oleh Stock Bot



2. Long Short-term Memory (LSTM)

- **LSTM** adalah singkatan dari **Long-Short Term Memory**. LSTM adalah jenis jaringan saraf berulang tetapi lebih baik daripada jaringan saraf berulang tradisional dalam hal memori.
- Memiliki pegangan yang baik dalam menghafal pola-pola tertentu LSTM berkinerja lebih baik. Seperti setiap NN lainnya, LSTM dapat memiliki beberapa lapisan tersembunyi dan saat melewati setiap lapisan, informasi yang relevan disimpan dan semua informasi yang tidak relevan dibuang di setiap sel.
- LSTM punya 3 gerbang utama: **FORGET** gate, **INPUT** gate, dan **OUTPUT** gate

Metode yang Digunakan Oleh Stock Bot



2. Long Short-term Memory (LSTM)

1. CELL Gate

Sebagai memori dari jaringan. Berfungsi sebagai jalan transportasi yang mentransfer informasi yang relevan selama proses sekuens.

2. FORGET Gate

Gerbang ini bertanggung jawab untuk memutuskan informasi mana yang disimpan untuk menghitung status sel dan mana yang tidak relevan dan dapat dibuang. Jika nilainya mendekati 0 yang berarti untuk dilupakan sedangkan jika mendekati 1 artinya disimpan

Metode yang Digunakan Oleh Stock Bot



2. Long Short-term Memory (LSTM)

3. INPUT Gate

Gerbang Input memperbarui status sel dan memutuskan informasi mana yang penting dan mana yang tidak. Seperti lupa gerbang membantu membuang informasi, gerbang masukan membantu untuk mengetahui informasi penting dan menyimpan data tertentu dalam memori yang relevan.

4. OUTPUT Gate

Gerbang Output menentukan next hidden state selanjutnya.

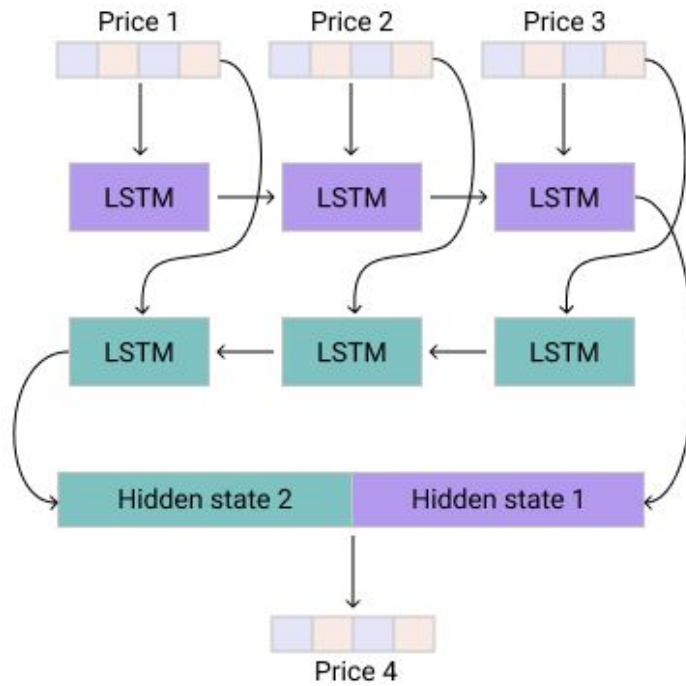
Pelatihan dan pengujian arsitektur model yang berbeda

Setelah menyiapkan kumpulan data, sekarang kita dapat mulai melakukan hal-hal menyenangkan — melatih berbagai model pembelajaran mendalam. Saya telah bereksperimen dengan Bidirectional LSTMs, Transformers, dan CNN + Bi-LSTM kode untuk semua model dapat ditemukan di Github.

Bidirectional-LSTM (Bi-LSTM)

LSTM Bidirectional adalah perpanjangan dari LSTM (Long Short Term Memory Cell) tradisional yang dapat meningkatkan kinerja model saat memproses kumpulan data sekuensial. LSTM dua arah menggabungkan dua lapisan LSTM individu sedangkan lapisan LSTM pertama menerima data berurutan (misalnya harga IBM) dalam urutan kronologis yang benar dan lapisan LSTM kedua menerima data yang sama hanya dalam urutan terbalik. Setelah data input diproses oleh lapisan LSTM dua arah (2 lapisan LSTM), kedua output digabungkan untuk menghasilkan prediksi akhir.

Bi-LSTM Network



Evaluasi Model

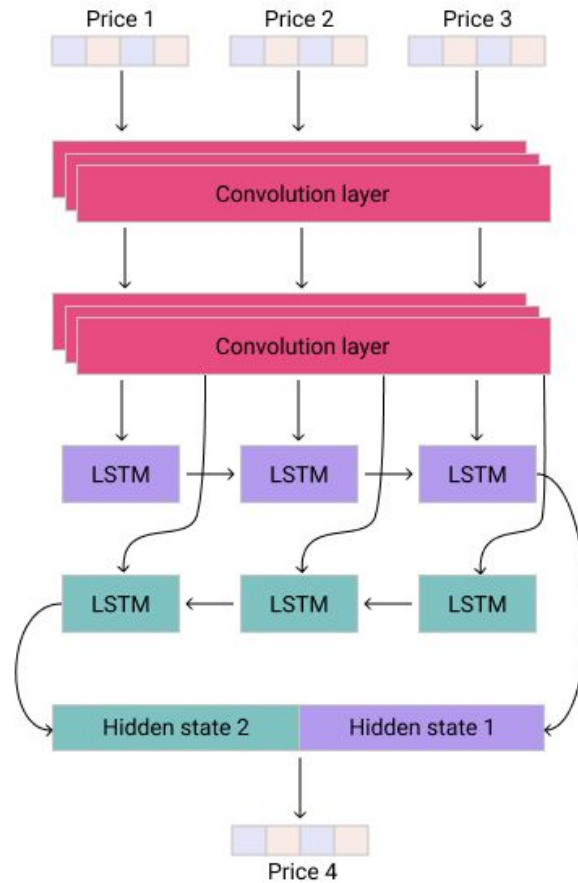
Setelah berlatih selama 200 zaman, diperoleh hasil berikut. Untuk set validasi, dimiliki Mean Average Percentage Error (MAPE) 3,5828 yang dapat ditafsirkan, demi kesederhanaan, sebagai akurasi 96,42%. Dataset pengujian memiliki nilai MAPE 4,2656, setara dengan akurasi 95,73%.

CNN + Bidirectional LSTM

Biasanya, CNN digunakan untuk klasifikasi gambar sedangkan setiap lapisan konvolusi dalam CNN mengekstraksi fitur yang berbeda dari gambar. Namun, dalam beberapa tahun terakhir telah ditunjukkan bahwa CNN memberikan nilai ketika menganalisis deret waktu dan data sekuensial (misalnya suara dan teks). Lapisan convolutional baik dalam mendeteksi pola yang terjadi antara titik data yang secara spasial berdekatan satu sama lain. Dalam hal data harga dan volume hari demi hari Anda seharusnya demikian.

Arsitektur lapisan konvensional telah terinspirasi oleh blok Inception Google. Mengubah konvolusi 2D dari model awal menjadi konvolusi 1D yang membuat lapisan kompatibel dengan deret waktu.

CNN + Bi-LSTM Network



Evaluasi Model



- Karena prediksi harga saham pada dasarnya adalah masalah regresi, **RMSE (Root Mean Squared Error)** dan **MAPE (Mean Absolute Percentage Error %)** akan menjadi metrik evaluasi model saat ini. Keduanya merupakan ukuran akurasi ramalan yang berguna.
- Formulasnya sbg berikut, dimana **N = jumlah titik waktu**, **At = harga saham aktual/sebenarnya**, **Ft = nilai prediksi/perkiraan**.

$$MAPE = \frac{1}{N} * \sum_{t=1}^N \left| \frac{At - Ft}{At} \right| \quad RMSE = \sqrt{\frac{1}{N} * \sum_{t=1}^N (At - Ft)^2}$$

- RMSE memberikan perbedaan antara nilai prediksi dan nilai sebenarnya, sedangkan MAPE (%) mengukur perbedaan ini relatif terhadap nilai sebenarnya. Misalnya, nilai MAPE 12% menunjukkan bahwa perbedaan rata-rata antara harga saham yang diprediksi dan harga saham sebenarnya adalah 12%.
- Makin kecil nilai MAPE dan/atau RMSE, maka semakin baik model

Mengambil Data Stock Real Time



```
[ ] pip install mplfinance
```

```
[ ] import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
import mplfinance as mpf
plt.style.use('seaborn')

import yfinance as yf
```

```
[ ] #mengakses data dari perusahaan di website yahoo finance
security = yf.Ticker('amzn')
# 'amzn' = kode saham perusahaan Amazon

stockinfo = security.info

#for key, value in stockinfo.items():
# print(key, ":", value)
```

Mengambil Data Stock Real Time



```
#memanggil tanggal hari ini dalam bentuk string
today = datetime.now().date().strftime("%Y-%m-%d")

#Mengambil data Historis pergerakan harga saham penutupan
DF = security.history(start = '2015-03-06', end= today)
# max artinya data historis harga saham full
# 1mo artinya data historis harga saham selama sebulan terakhir, untuk data full gunakan 'max'
# bisa juga menspesifikasi tanggal yyyy-mm-dd untuk start dan end
DF
```

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2015-03-06	385.519989	387.000000	378.880005	380.089996	2627000	0	0
2015-03-09	378.399994	379.329987	375.279999	378.559998	2311400	0	0
2015-03-10	377.450012	377.769989	369.179993	369.510010	3121000	0	0
2015-03-11	370.609985	373.350006	366.260010	366.369995	2495100	0	0
2015-03-12	368.820007	375.500000	367.519989	374.239990	2810000	0	0
...
2022-02-28	3048.500000	3089.000000	3017.000000	3071.260010	2878500	0	0
2022-03-01	3054.649902	3081.979980	2999.540039	3022.840088	2243700	0	0
2022-03-02	3016.979980	3059.989990	2975.000000	3041.050049	2366700	0	0
2022-03-03	3070.629883	3078.580078	2937.120117	2957.969971	3259900	0	0
2022-03-04	2943.179932	2957.000000	2876.139893	2912.820068	3046700	0	0

1763 rows × 7 columns

Program Python untuk SMA



▼ Simple Moving Average

Masih menggunakan data Saham di section sebelumnya

```
[ ] DF.head()
```

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2015-03-06	385.519989	387.000000	378.880005	380.089996	2627000	0	0
2015-03-09	378.399994	379.329987	375.279999	378.559998	2311400	0	0
2015-03-10	377.450012	377.769989	369.179993	369.510010	3121000	0	0
2015-03-11	370.609985	373.350006	366.260010	366.369995	2495100	0	0
2015-03-12	368.820007	375.500000	367.519989	374.239990	2810000	0	0

```
[ ] moving_average_window = 20
```

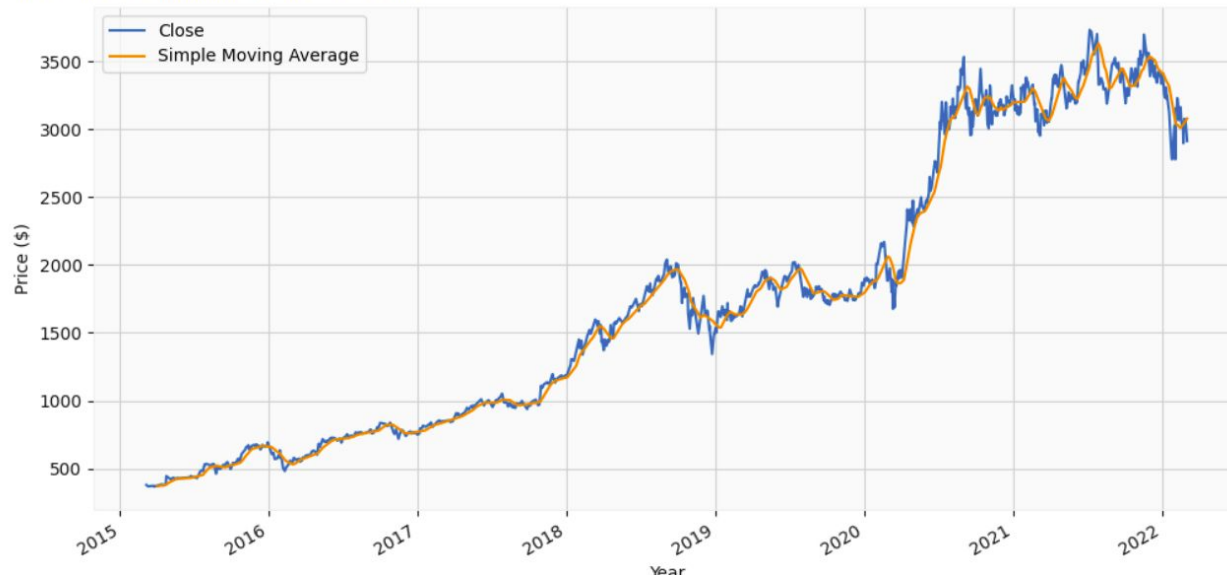
```
[ ] moving_average_pandas = DF['Close'].rolling(window = moving_average_window).mean()  
moving_average_pandas
```

Perbandingan Training Model



SMA vs Real

<matplotlib.legend.Legend at 0x7f388e2cfdd0>



```
plt.figure(figsize = (12,6))
plt.plot(DF['Close'], label = "Close")
moving_average_pandas.plot(label = "Simple Moving Average")
plt.xlabel('Year')
plt.ylabel('Price ($)')
plt.legend()
```

Program Python untuk LSTM



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pandas_datareader as web
import yfinance as yf
import datetime as dt

from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM

#untuk load data saham
company='FB'#atau AMZN

start=dt.datetime(2012,1,1)
end=dt.date.today()-dt.timedelta(days=5)#kalau error cobain 2022,3,5

data = yf.download(company, start,end)
```

```
#mempersiapkan datanya
scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(data["Close"].values.reshape(-1,1))

prediction_days=60

x_train=[]
y_train=[]

for x in range(prediction_days, len(scaled_data)):
    x_train.append(scaled_data[x-prediction_days:x, 0])
    y_train.append(scaled_data[x, 0])

x_train, y_train= np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1],1))
```

Program Python untuk LSTM



```
#membuat model

model=Sequential()

model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(Dropout(0.2))

model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))

model.add(LSTM(units=50))
model.add(Dropout(0.2))

model.add(Dense(units=1))#predictio untuk next closenya

model.compile(optimizer="adam", loss="mean_squared_error")
model.fit(x_train, y_train, epochs=25, batch_size=32)#nanti epochs balikin ke 25

#test model akurasi dari data yang dipunya
```

```
#load test datanya
test_start=dt.datetime.today()-dt.timedelta(days=5)
test_end=dt.datetime.today()

test_data=yf.download(company, start,end)
actual_prices=test_data["Close"].values

total_dataset=pd.concat((data["Close"], test_data["Close"]), axis=0)

model_inputs = total_dataset[len(total_dataset) - len(test_data) - prediction_days:].values
model_inputs = model_inputs.reshape(-1,1)
model_inputs = scaler.transform(model_inputs)

#prediction dari test data
x_test = []

for x in range(prediction_days, len(model_inputs)):
    x_test.append(model_inputs[x-prediction_days:x,0])

x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

predicted_prices = model.predict(x_test)
predicted_prices = scaler.inverse_transform(predicted_prices)
```

Program Python untuk LSTM



```
#Plotting test prediksinya
plt.figure(figsize=(10,10))
plt.plot(actual_prices, color="green", label=f"Actual {company} Price")
plt.plot(predicted_prices, color="pink", label=f"Predicted {company} Price")
plt.title(f"{company} Share Price")
plt.xlabel("Time")
plt.ylabel(f"{company} Share Price")
plt.legend()
plt.show()

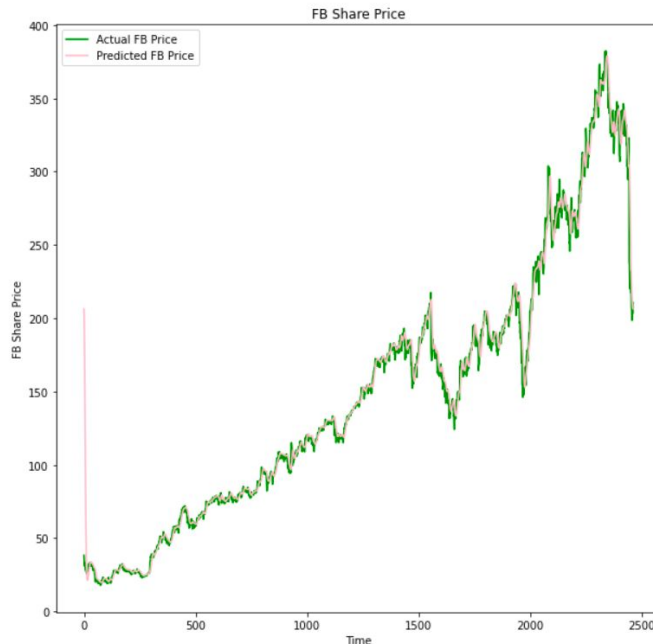
#Memprediksi saham hari esok
real_data = [model_inputs[len(model_inputs) - prediction_days:len(model_inputs+1), 0]]
real_data = np.array(real_data)
real_data = np.reshape(real_data, (real_data.shape[0], real_data.shape[1], 1))

prediction = model.predict(real_data)
prediction = scaler.inverse_transform(prediction)
print(f"ini adalah hasil prediksinya {prediction}")
```


Perbandingan Training Model



LSTM vs Real



```
plt.figure(figsize=(10,10))
plt.plot(actual_prices, color="green", label=f"Actual {company} Price")
plt.plot(predicted_prices, color="pink", label=f"Predicted {company} Price")
plt.title(f"{company} Share Price")
plt.xlabel("Time")
plt.ylabel(f"{company} Share Price")
plt.legend()
plt.show()
```