

LAPORAN AKHIR PROYEK PEMODELAN MATEMATIS
“STOCK TRADING ROBOT”



Disusun Oleh :

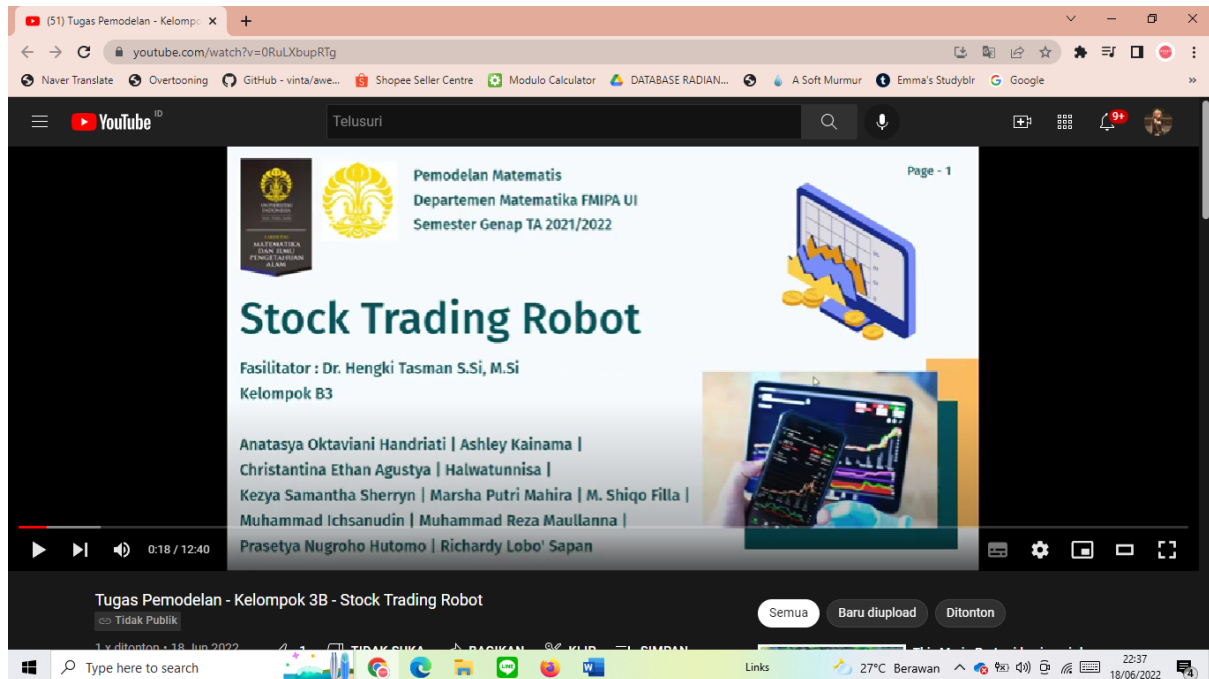
Anatasya Oktaviani Handriati	(1906296186)
Ashley Kainama	(1906352073)
Christantina Ethan Agustya	(1906351921)
Halwatunnisa	(1906373960)
Kezya Samantha Sherryn	(1906296280)
Marsha Putri Mahira	(1906296293)
M. Shiqo Filla	(1906352016)
Muhammad Ichsanudin	(1906351940)
Muhammad Reza Maullanna	(1906373916)
Prasetya Nugroho Hutomo	(1906307353)
Richardy Lobo' Sapan	(1906373954)

Dosen Pembimbing: Dr. Hengki Tasman S.Si, M.Si.

DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS INDONESIA
DEPOK
2022

LINK VIDEO: <https://www.youtube.com/watch?v=0RuLXbupRTg>

SCREENSHOT:



KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Terima kasih atas nikmat Tuhan Yang Maha Esa. yang begitu banyak rahmat serta karunia-Nya sehingga penulis bisa menyelesaikan draft laporan akhir mengenai “*Trading Robot*”. Selain itu, tim penulis turut mengucapkan rasa terima kasih terhadap Dr. Hengki Tasman S.Si, M.Si., selaku dosen pembimbing mata kuliah Pemodelan Matematis.

Tujuan dari penulisan makalah ini yaitu sebagai syarat dalam memenuhi tugas mata kuliah Pemodelan Matematis. Pada draft laporan akhir ini, tim penulis akan menjelaskan mengenai robot trading dalam perangkat lunak Python.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Depok, 23 Mei 2022

Tim Penulis

DAFTAR ISI

KATA PENGANTAR	3
DAFTAR ISI	4
ABSTRAK	6
BAB I	7
PENDAHULUAN	7
1.1. Latar Belakang	7
1.2. Uraian singkat proyek	7
1.3. Rumusan Masalah	7
1.4. Tujuan Penelitian	7
1.5. Manfaat Penelitian	8
1.6. Metode Studi Pustaka	8
BAB II	9
TELAAH PUSTAKA	9
2.1 Saham dan Konsep Dasarnya	9
2.1.1 Pengertian	9
2.1.2 Jenis-Jenis Saham	9
2.1.3 Berdasarkan kemampuannya:	10
2.1.4 Alur Transaksi Saham	10
2.1.5 Cara Memilih Saham	11
2.2 Stock Trading Robot	12
2.2.1 Pengertian	12
2.2.2 Peran	12
2.3 Beberapa Metode dalam Stock Trading Robot	12
2.3.1 ARIMA	12
2.3.2 LSTM	14
2.4 Metrik Evaluasi	14
BAB III	16
3.1 Pemodelan Stock Trading Robot	16
3.1.1 Asumsi	16
3.1.2 Batasan Masalah	16
3.1.3 Metode yang Digunakan dalam Proyek Stock Trading Robot	17
3.2 Pemodelan dan Program Stock Trading Robot	18
3.2.1 Perpajakan dalam Trading Saham	18
3.2.2 Underfitting dan Overfitting	19
3.2.3 Kriteria Jual, Beli, dan Tahan	19
3.3 Simulasi dan Output Program	20

3.3.1 Persiapan	20
3.3.3 Output dari Program	21
3.4 Analisis lainnya	26
3.4.1 Trial and Error dalam penentuan nilai Train:Test	26
3.4.2 Up-down kebenaran forecasting aproksimasi vs data eksak	26
3.4.3 Asumsi Besarnya Error pada Analisis Error Saham Unilever	28
BAB IV	30
PENUTUPAN	30
4.1 Kesimpulan	30
4.2 Saran	30
DAFTAR PUSTAKA	31
LAMPIRAN	33

ABSTRAK

Masa kini, saham menjadi salah satu investasi yang diminati karena menawarkan keuntungan jangka panjang. Cukup banyak orang yang berani berkecimpung dalam dunia trading saham, walau ia memiliki pengetahuan yang sedikit mengenai saham. Tidak sedikit dari mereka yang mengalami kerugian akibat asal membeli atau menjual saham tanpa teori yang mumpuni. Karenanya, diciptakan suatu teknologi, dalam hal ini robot, yang dapat membantu manusia mengatasi masalah ini. Walau teknologi ini sangat membantu, ada pihak-pihak yang tidak bertanggung jawab yang meraup keuntungan besar dengan bisnis robot trading yang bodong. Untuk itu penelitian ini dilakukan untuk dapat mengetahui dan membuat robot *trading*. Penelitian ini menggunakan penerapan metode LSTM pada perangkat lunak Python. Robot *trading* dapat memprediksi harga saham hingga sepuluh hari kedepan dan memberikan rekomendasi tindakan apa yang diambil oleh pengguna. Tindakan yang direkomendasikan adalah beli, jual, atau tahan.

Kata kunci: Trading; LSTM; RMSE; BBKA.JK; UNVR.JK; SIDO.JK

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dengan 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 terjadinya kesalahan pada proses perdagangan saham di Indonesia. Tentunya saham mempunyai harga yang fluktuatif yang sekilas bergerak secara acak. Untuk membuat sebuah robot trading, dibutuhkan suatu metode untuk dapat memprediksi pergerakan harga saham. Selain metode *forecasting* yang tepat, dalam membuat sebuah robot trading, dibutuhkan juga algoritma jual-beli yang sesuai. Karena harga saham yang fluktuatif, maka sangat penting untuk memperhatikan waktu saat membeli atau menjual saham.

1.2. Uraian singkat proyek

Menerapkan metode LSTM dalam perangkat lunak Python sehingga didapat prediksi harga saham yang dimanfaatkan untuk pengambilan keputusan membeli atau menjual saham yang dimiliki.

1.3. Rumusan Masalah

Berikut rumusan masalah dalam penelitian yang kami buat

- Metode apa yang lebih baik digunakan oleh Stock Trading Robot dalam memprediksi harga saham?
- Bagaimana cara memvisualisasikan hasil forecasting harga saham dengan metode tersebut?
- Bagaimana kriteria yang perlu Stock Trading Robot pertimbangkan dalam melakukan jual/beli suatu saham?

1.4. Tujuan Penelitian

Berikut tujuan penelitian berdasarkan rumusan masalah sebelumnya

1. Menentukan metode yang lebih baik digunakan oleh Stock Trading Robot dalam memprediksi harga saham

2. Memperlihatkan hasil forecasting harga saham dengan metode tersebut.
3. Menentukan kriteria dalam jual/beli saham oleh Stock Trading Robot

1.5. Manfaat Penelitian

Penelitian ini bertujuan untuk mengembangkan sebuah trading robot berbasis LSTM yang dapat memprediksi harga saham serta membuat keputusan jual-beli sederhana seperti yang tertera di 2.3.6 . Diharapkan pengembangan robot ini dapat memberikan suatu derajat otomasi bagi perdagangan saham, dan meminimalkan kerugian yang biasanya didapat pemula

1.6. Metode Studi Pustaka

Studi pustaka untuk makalah ini dibagi jadi 2 bagian:

- a. Studi pustaka saham dan konsep dasarnya (2.1)
- b. Studi pustaka metode yang dipakai untuk robot trading (2.3)

Dimana untuk bagian a. merujuk pada buku tulisan Victor purba, “Perkembangan dan Struktur Pasar Modal Indonesia Menuju Era AFTA 2003”. Dan untuk bagian b. kami merujuk pada bermacam artikel yang anggota cari masing-masing di internet (untuk daftar artikel selengkapnya ada di daftar pustaka)

BAB II

TELAAH PUSTAKA

2.1 Saham dan Konsep Dasarnya

2.1.1 Pengertian

Menurut Sapto Raharjo (2004), saham merupakan surat berharga yang isinya adalah bukti kepemilikan atau sebuah pernyataan dari seseorang atau instansi perusahaan. Sedangkan menurut Tjiptono Darmadji dan Hendy M. Fakhruddin (2002), saham adalah surat bukti atau tanda bukti suatu kepemilikan bagi individu atau instansi pada perusahaan perseroan terbatas. Oleh karena itu, saham adalah suatu alat bukti atas kepemilikan dari sebuah perusahaan yang berupa lembaran-lembaran surat berharga.

2.1.2 Jenis-Jenis Saham

Dalam pasar uang, ada beragam jenis saham. Jenis jenis saham menurut para ahli terbagi berdasarkan sektor dan berdasarkan berdasarkan kemampuannya. Berikut adalah penjelasannya.

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.

2.1.3 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)

2. Saham Preferen (*Preferred Stock*)

Ciri-ciri:

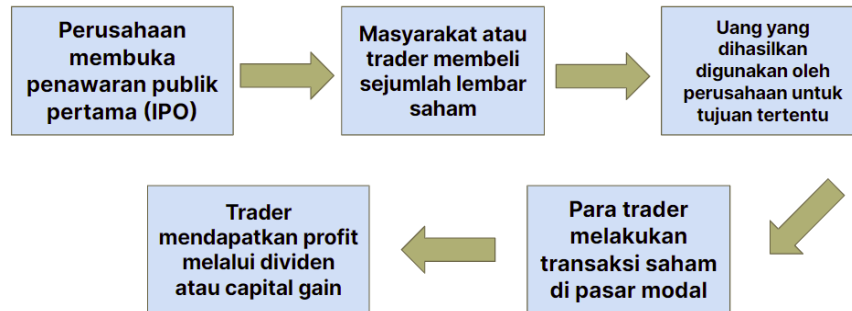
- ❖ Memiliki beberapa tingkatan dengan karakteristik yang berbeda
- ❖ Ada tagihan atas suatu pendapatan, memiliki prioritas tinggi dalam hal pembagian nilai dividen
- ❖ Bisa ditukarkan dengan saham biasa, dengan kesepakatan yang ada

2.1.4 Alur Transaksi Saham

Pelaksanaan perdagangan Efek di Bursa dilakukan dengan menggunakan fasilitas JATS NEXT-G. Perdagangan Efek di Bursa hanya dapat dilakukan oleh Anggota Bursa (AB) yang juga menjadi Anggota Kliring KPEI. Anggota Bursa Efek bertanggung jawab terhadap seluruh transaksi yang dilakukan di Bursa baik untuk kepentingan sendiri maupun untuk kepentingan nasabah.

Alur transaksi saham bisa dilihat pada gambar di bawah ini:

Alur Transaksi Saham



Gambar 1: Alur transaksi saham

Alur ini diawali dengan perusahaan yang ingin membuka penawaran publik pertama/ initial public offering. Setelah itu, masyarakat atau trader secara khusus dapat membeli sejumlah besar lembar saham. Uang yang dihasilkan akan digunakan oleh perusahaan tersebut untuk pendanaan perusahaan mereka. Trader dapat melakukan transaksi saham pada tahap ini. Keuntungan dari trader bisa berasal dari capital gain ataupun dividen dari perusahaan.

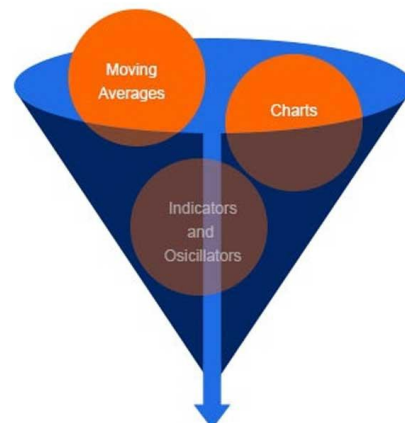
2.1.5 Cara Memilih Saham

Fundamental Analysis



VS

Technical Analysis



Gambar 2: Analisis dalam Memilih Saham

Setelah mengetahui bagaimana saham bekerja, penting juga untuk mengerti tentang cara memilih saham. Terdapat dua analisis yang digunakan untuk mengambil keputusan dalam transaksi saham. Pertama disebut *Fundamental Analysis*. *Fundamental Analysis* merupakan analisis yang bertujuan mempelajari hal-hal yang berhubungan dengan kondisi keuangan perusahaan tertentu. Selanjutnya *Technical Analysis*, yaitu analisis yang berdasarkan pada data harga historis yang terjadi pada pasar saham. Sesuai namanya, analisa teknikal melibatkan pendekatan teknis dengan penggunaan dua metode. Kedua metode itu adalah metode klasik (grafik harga saham) dan metode modern (indikator dengan algoritma atau rumus statistik).

2.2 Stock Trading Robot

2.2.1 Pengertian

Sistem perdagangan otomatis (trading robot) merupakan sistem memungkinkan pengguna untuk menetapkan aturan khusus untuk data yang diberikan pada sistem dan memberikan hasil secara otomatis setelah melalui proses dalam komputer.

2.2.2 Peran

Perdagangan saham bukanlah hal yang mudah, dan gampang untuk trader terjerumus dalam hal-hal simpel. Sekiranya trading robot yang akan dibuat guna membantu pemegang saham dengan mengotomasikan peran-peran berikut:

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

2.3 Beberapa Metode dalam Stock Trading Robot

2.3.1 ARIMA

Arima merupakan gabungan dari 3 model utama yaitu: *Autoregressive* (AR), *Integrated* (I), dan *Moving Average* (MA) yang digabung menjadi satu keutuhan metode yang dikenal sebagai model ARIMA. Untuk komponen-komponenya adalah:

- Model Auto Regressive (hanya AR) murni adalah model di mana Y_t hanya bergantung pada lag-nya sendiri. Artinya, **Y_t adalah fungsi dari 'lag dari Y_t '**.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p} + \epsilon_t$$

- **Integrated** di sini bermakna **mengurangi nilai saat ini dari sebelumnya** dan dapat digunakan untuk mengubah deret waktu menjadi **deret waktu yang stasioner**.

- Differencing **orde pertama** -> tren linier -> transformasi $z_i = y_i - y_{i-1}$.
- Differencing **orde kedua** -> tren kuadrat -> $z_i = (y_i - y_{i-1}) - (y_{i-1} - y_{i-2})$,
- dan seterusnya.

- Demikian pula model Moving Average (hanya MA) adalah model di mana **Yt hanya bergantung pada error perkiraan yang lag**

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \dots + \phi_q \epsilon_{t-q}$$

- Model ARIMA adalah model di mana **deret waktu di-differencing setidaknya sekali ($d \geq 1$) untuk membuatnya stasioner dan menggabungkan suku-suku AR dan MA**. Sehingga persamaannya menjadi:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p} + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \dots + \phi_q \epsilon_{t-q}$$

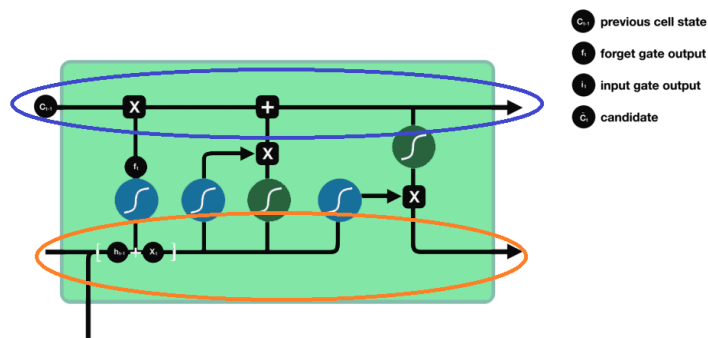
Prediksi Y_t = **Konstanta + Kombinasi linier Lag dari Y (hingga p lag) + Kombinasi Linier dari kesalahan prakiraan Lagged (hingga q lag)**

Tiga bilangan bulat (p, d, q) biasanya digunakan untuk parametrize model ARIMA.

- **p**: jumlah suku autoregresif (order AR)
- **d**: jumlah differencing (order difference)
- **q**: jumlah suku Moving Average (order MA)

2.3.2 LSTM

LSTM adalah singkatan dari **Long-Short Term Memory** yang mana adalah suatu pendekatan neural networking berulang. **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 4 gerbang utama: **FORGET** gate, **INPUT** gate, **OUTPUT** gate, dan **Cell** gate



Gambar 3: Alur Kerja Metode *Long-Short Term Memory* (LSTM)

Cara Kerja

- Pada Metode LSTM, bagian **Short term** adalah data yang diambil pada proses iterasi pembelajaran data oleh AI yang nantinya akan dinilai oleh AI apakah data tersebut memiliki korelasi dengan data yang lainnya.
- Sedangkan **Long Term** pada LSTM, adalah hasil pembelajaran yang terakumulasi (hidden cell) yang menentukan apakah data dari short term dapat diteruskan untuk diakumulasikan atau tidak.

2.4 Metrik Evaluasi

Karena prediksi harga saham pada dasarnya adalah masalah regresi, kita bisa menggunakan RMSE (Root Mean Squared Error) untuk menghitung tingkat akurasi model.

Formulanya sebagai berikut, dimana N = jumlah titik waktu, A_t = harga saham aktual/sebenarnya, F_t = nilai prediksi/perkiraan.

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

RMSE memberikan selisih antara nilai prediksi dan nilai sebenarnya. Makin kecil nilai RMSE, maka semakin baik model

BAB III

ANALISIS DAN SINTESIS

3.1 Pemodelan *Stock Trading Robot*

3.1.1 Asumsi

Dalam memodelkan masalah ini, kita memerlukan beberapa asumsi untuk membatasi permasalahan yang kita buat. Berikut adalah asumsi yang diperlukan:

1. Harga saham dipengaruhi oleh harga historisnya
2. Harga saham yang digunakan adalah harga penutupan
3. Keadaan suatu perusahaan akan terefleksi pada keadaan sahamnya (*technical analysis*)
4. Saham dari perusahaan sifatnya sah (bukan investasi bodong, dkk) terdaftar pada Bursa Efek Indonesia/Yahoo Finance
5. Harga saham dipengaruhi oleh harga historisnya
6. Harga saham yang digunakan adalah harga penutupan
7. Keadaan suatu perusahaan akan terefleksi pada keadaan sahamnya (*technical analysis*)
8. Saham dari perusahaan sifatnya sah (bukan investasi bodong atau sejenisnya) atau dengan kata lain terdaftar pada Bursa Efek Indonesia
9. Program memprediksi harga saham 10 hari ke depan
10. Biaya lain-lain ketika membeli: 0.15% (broker fee) + 0.04% (biaya transaksi) = 0.19%
11. Biaya lain-lain ketika menjual: 0.25% (broker fee) + 0.04% (biaya transaksi) + 0.1% (pajak) = 0.39%
12. Pembagian data test train adalah Test:Train = 4:96 (karena keterbatasan perangkat keras)

3.1.2 Batasan Masalah

Selain batasan masalah, diperlukan juga batasan masalah untuk model yang akan dibuat. Berikut adalah batasan untuk model tersebut:

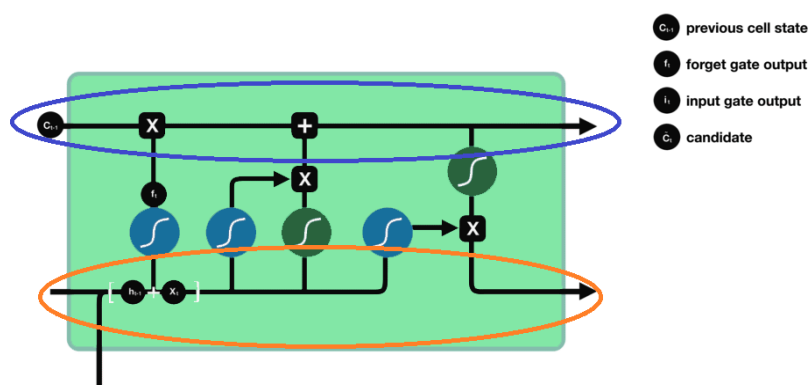
1. Periode pengecekan setiap hari penutupan

2. Contoh diambil 3 saham penutupan dari perusahaan berbeda (BCA JKT, SIDOMUNCUL JKT, UNILEVER JKT)
3. Data diambil 10 tahun terakhir
4. Hari libur tidak dihitung pada pengambilan *data close*
5. Periode pengecekan setiap hari
6. Contoh diambil 3 (tiga) saham penutupan dari perusahaan berbeda (BCA JKT, SIDOMUNCUL JKT, UNILEVER JKT)
7. Hari libur tidak dihitung pada pengambilan *data close*
8. Data diambil 10 (sepuluh) tahun terakhir

3.1.3 Metode yang Digunakan dalam Proyek *Stock Trading Robot*

ong short term memory network (LSTM) adalah salah satu modifikasi dari recurrent neural network atau RNN. Banyak modifikasi dari RNN, tetapi LSTM merupakan salah satu yang populer di antaranya. LSTM hadir untuk melengkapi kekurangan RNN yang tidak dapat memprediksi kata berdasarkan informasi lampau yang disimpan dalam jangka waktu lama. Dengan demikian, LSTM mampu mengingat kumpulan informasi yang telah disimpan dalam jangka waktu panjang, sekaligus menghapus informasi yang tidak lagi relevan. LSTM lebih efisien dalam memproses, memprediksi, sekaligus mengklasifikasikan data berdasarkan urutan waktu tertentu.

Metode yang digunakan dalam *trading robot* ini adalah Long Short-term Memory (LSTM). LSTM punya 4 gerbang utama: FORGET gate, INPUT gate, OUTPUT gate, dan Cell gate



Gambar 4: Cara Kerja Metode LSTM

Cara Kerja

- Pada Metode LSTM, bagian **Short term** adalah data yang diambil pada proses

iterasi pembelajaran data oleh AI yang nantinya akan dinilai oleh AI apakah data tersebut memiliki korelasi dengan data yang lainnya.

- Sedangkan **Long Term** pada LSTM, adalah hasil pembelajaran yang terakumulasi (hidden cell) yang menentukan apakah data dari short term dapat diteruskan untuk diakumulasikan atau tidak.

3.2 Pemodelan dan Program Stock Trading Robot

3.2.1 Perpajakan dalam Trading Saham

Kewajiban perpajakan akan muncul apabila seorang investor mendapatkan dividen. Sebab, pajak yang dikenakan merupakan jenis pajak penghasilan. Sekedar mengingatkan, penting untuk memperhatikan kembali bahwa pajak penjualan saham bersifat final. Dengan demikian, pajak tersebut akan dipotong secara otomatis oleh petugas pajak. Pajak penjualan saham berlaku bagi setiap transaksi penjualan saham. Seorang investor tetap perlu menyetor pajak meski melakukan jual rugi. Sebab, bagaimanapun juga, pajak ini tidak memperhatikan laba atau rugi investor dari penjualan yang mereka lakukan. Berikut adalah rincian untuk pajak dan biaya dalam transaksi saham

Tiga biaya yang dikenakan pada trader saat melakukan transaksi saham

- Pajak penjualan saham (0.1%)
- Biaya transaksi (0.04%)
- Biaya broker (0.10%-0.35%)

Biaya saat membeli:

- Biaya transaksi: 0.04%
 - Biaya broker: 0.15%
- Total: 0.19%

Biaya ini selanjutnya akan disebut ongkos beli.

Biaya saat menjual:

- Biaya transaksi: 0.04%
- Pajak: 0.1%
- Biaya broker: 0.25%

Total: 0.39%

Biaya ini selanjutnya akan disebut ongkos jual.

3.2.2 Underfitting dan Overfitting

Overfitting dan Underfitting merupakan keadaan dimana terjadi defisiensi yang dialami oleh kinerja model machine learning. Salah satu fungsi utama dari machine learning adalah untuk melakukan generalisasi dengan baik, terjadinya overfitting dan underfitting menyebabkan machine learning tidak dapat mencapai salah satu tujuan utamanya, yaitu generalisasi. Generalisasi disini adalah mencari tren dominan dari suatu data untuk menghasilkan insight guna memprediksi data yang belum pernah dilihat dengan tepat. Keduanya merupakan masalah fundamental yang kerap kali dialami bahkan oleh seorang ahli.

Underfitting terjadi ketika model tidak bisa melihat logika dibelakang data, hingga tidak bisa melakukan prediksi dengan tepat, baik untuk dataset training maupun dataset lain yang serupa. Underfitting model akan memiliki *high training error* dan *high test error*.

Untuk mengatasi ini ada beberapa hal yang bisa dilakukan, diantaranya:

- Menambah jumlah parameter ML model
- Menambah complexity atau tipe dari model
- Menambah training time sampai cost function minimal

Overfitting terjadi karena model yang dibuat terlalu fokus pada training dataset tertentu, hingga tidak bisa melakukan prediksi dengan tepat jika diberikan dataset lain yang serupa. Overfitting biasanya akan menangkap data noise yang seharusnya diabaikan. overfitting model akan memiliki *low training error* dan *high test error*.

Untuk mengatasi ini ada beberapa hal yang bisa dilakukan, diantaranya:

- Data Augmentation
- Generalization

3.2.3 Kriteria Jual, Beli, dan Tahan

Untuk memungkinkan model menganalisa dan membuat keputusan, diperlukan suatu kriteria yang menentukan apakah saham akan dibeli, dijual, atau ditahan. berikut adalah kriteria tersebut:

1. Kriteria beli:

Jika tidak punya saham atau ingin menambah kuantitas saham dan $HS_{n+1} > HS_n$, maka beli pada hari ke n

2. Kriteria tahan: (Asumsi sudah punya saham)

- Jika $HS_{n+1} \geq HS_n$, maka Saham ditahan atau
- Jika HJ setelah dikurangi ongkos jual $<$ HB setelah ditambah dengan ongkos beli, maka saham ditahan

3. Kriteria jual: (Asumsi sudah punya saham)

Jika $HS_n - HS_{n+1} > 0$ dan Jika HJ setelah dikurangi ongkos jual $>$ dari HB setelah ditambah ongkos beli, maka saham dijual di hari ke n

Untuk $n \in \mathbb{N}$, $n = 0, 1, 2, 3, \dots, 10$

Asumsi kita berada pada hari ke-0

dengan n adalah bilangan terkecil yang memenuhi

Keterangan:

HS = Harga Saham, HS_0 : harga eksak di saat trader sedang menjalankan program

HJ = Harga Jual

HB = Harga Beli

3.3 Simulasi dan Output Program

3.3.1 Persiapan

Beberapa persiapan jika ini adalah pertama kalinya ingin menggunakan program *stock robot trading*:

1. Sebelum memulai proses input, pastikan terlebih dahulu python sudah memiliki modul/library untuk mengakses data di *yahoo finance*. Jika python tidak memiliki modul, maka instal library tersebut dengan menjalankan code:
`pip install yfinance --upgrade --no-cache-dir`
pada cell python.

```
[ ] pip install yfinance --upgrade --no-cache-dir

Collecting yfinance
  Downloading yfinance-0.1.70-py2.py3-none-any.whl (26 kB)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (from yfinance) (1.21.5)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.7/dist-packages (from yfinance) (0.0.10)
Collecting lxml>=4.5.1
  Downloading lxml-4.8.0-cp37m-manylinux_2_17_x86_64_manylinux2014_x86_64_manylinux_2_24_x86_64.whl (6.4 MB)
  100% |#####| 6.4 MB 12.7 MB/s
Requirement already satisfied: pandas>=0.24.0 in /usr/local/lib/python3.7/dist-packages (from yfinance) (1.3.5)
Collecting requests>=2.26
  Downloading requests-2.27.1-py2.py3-none-any.whl (63 kB)
  100% |#####| 63 kB 28.1 MB/s
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.0->yfinance) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.0->yfinance) (2018.9)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>=0.24.0->yfinance) (1.15.0)
Requirement already satisfied: charset-normalizer<=2.0.0 in /usr/local/lib/python3.7/dist-packages (from requests>=2.26->yfinance) (2.0.12)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.26->yfinance) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.26->yfinance) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.26->yfinance) (1.24.3)
Installing collected packages: requests, lxml, yfinance
  Attempting uninstall: requests
    Found existing installation: requests 2.23.0
    Uninstalling requests-2.23.0:
      Successfully uninstalled requests-2.23.0
  Attempting uninstall: lxml
    Found existing installation: lxml 4.2.6
    Uninstalling lxml-4.2.6:
      Successfully uninstalled lxml-4.2.6
Successfully installed lxml-4.8.0 requests-2.27.1 yfinance-0.1.70
```

Gambar 5: Import Yahoo Finance

2. Kemudian tentukan dahulu data saham apa yang ingin digunakan dengan mengambil bagian *Ticker* dari suatu saham, seperti contoh: BBKA.JK, UNVR.JK, SIDO.JK. d.l.l.

3.3.2 Input dari Program

Untuk menjalankan program, dibutuhkan input berupa:

1. *Ticker* dari suatu saham, seperti: BBKA.JK, UNVR.JK, SIDO.JK (biasanya harus dalam kapital semua)
2. Status sedang tidak memiliki saham/sedang ingin menambah saham
3. Besarnya ongkos beli dan ongkos jual yang diinginkan

```
Masukan Ticker saham yang ingin dianalisis program, sebagai contoh (BBKA.JK): BBKA.JK
[*****100%*****] 1 of 1 completed
Apakah anda sedang 'tidak' memiliki saham atau 'ingin membeli' saham baru? Jawab dengan ya/tidak: ya
Apakah anda menginginkan 'ongkos beli' (pajak beli) berbeda (bawaan adalah 0.0019), jawab dengan ya/tidak: ya
Masukan besarnya 'ongkos beli' (pajak beli) baru dengan decimal: 0.0019
Apakah anda menginginkan 'ongkos jual' (pajak jual) berbeda (bawaan adalah 0.0039), jawab dengan ya/tidak: tidak
```

Gambar 6: Input program

3.3.3 Output dari Program

Selanjutnya akan ditunjukkan hasil output program. Beberapa hal yang perlu diperhatikan:

1. Nilai untung per lembar saham adalah Harga yang sudah terkena pajak beli dan jual.
2. Hari 0 adalah hari pertama kali trader mengakses program (hari prediksi ke 0) sehingga pada hari 0 nilai saham adalah nilai eksak. Untuk hari 1 dan seterusnya adalah harga prediksi.

Berikut adalah simulasi program dari saham SIDO.JK pada Yahoo!Finance yang dijalankan pada 24 Mei 2022 dengan asumsi trader belum memiliki saham di perusahaan tersebut.

```
[*****100%*****] 1 of 1 completed
Epoch 1/3
63/63 [=====] - 13s 128ms/step - loss: 0.0138
Epoch 2/3
63/63 [=====] - 8s 127ms/step - loss: 0.0028
Epoch 3/3
63/63 [=====] - 8s 126ms/step - loss: 0.0027
Maka hasil forecasting saham untuk ke 2022-05-25 adalah:
ini adalah hasil prediksinya [[885.91785]]
Dimana RMSEnya adalah: 47.928111946486474

960.0 harga saat hari ke: 0

[[885.91785]] harga saat hari ke: 1

Epoch 1/3
63/63 [=====] - 13s 125ms/step - loss: 0.0128
Epoch 2/3
63/63 [=====] - 8s 130ms/step - loss: 0.0023
Epoch 3/3
63/63 [=====] - 8s 127ms/step - loss: 0.0025
Maka hasil forecasting saham untuk ke 2022-05-26 adalah:
ini adalah hasil prediksinya [[921.95605]]
Dimana RMSEnya adalah: 40.36257965926213

[[885.91785]] harga saat hari ke: 1

[[921.95605]] harga saat hari ke: 2

Beli saham pada hari ke 1, karena besok akan naik

Harga saham yang dibeli adalah [[885.91785]]

Saham dalam keadaan baik dan meningkat untuk hari prediksi ke: 2

[[935.54877]] harga saat hari ke: 3

[[898.404]] harga saat hari ke: 4

Saham harus dijual saat hari: 3 karena besok akan jatuh

Keuntungan yang akan didapat persatuan sahamnya sebesar: [[44.299072]]
```

Gambar 7: Output program

Selanjutnya ditunjukkan hasil simulasi program dari ketiga perusahaan.

SIDO.JK

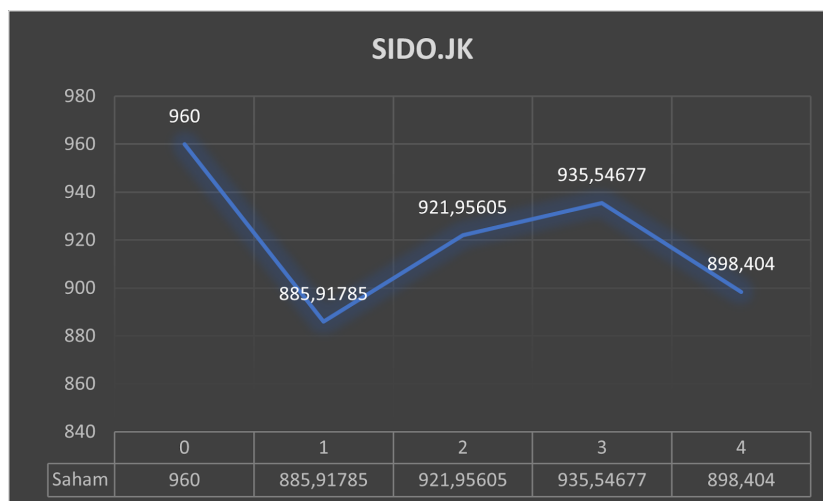
Hari	SIDO.JK	Status
0	960.0	Belum punya saham dan tidak beli
1	885.91785	Beli saham
2	921.95605	Tahan
3	935.54877	Jual (Untung = 44.299072)
4	898.404	

Tabel 1: Analisis Jual-Beli Saham SIDO.JK



(gambar di atas diambil dari program yang dijalankan pada 13/6/2022)

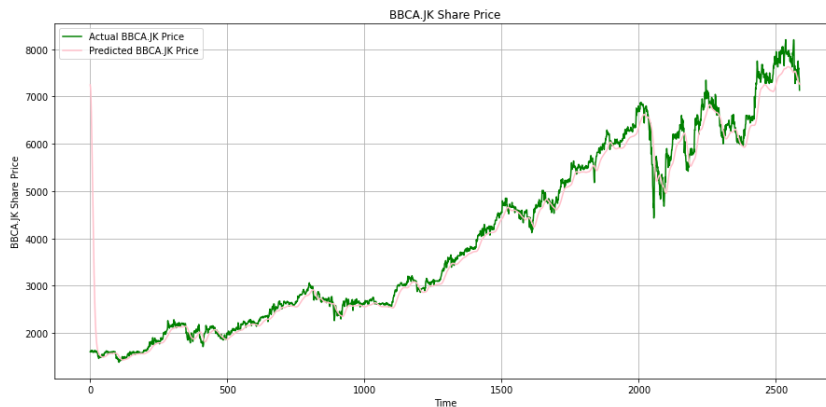
Gambar 8: Perbandingan Hasil Prediksi dan Harga Asli Saham SIDOJK



**Gambar 9: Hasil Forecasting Saham SIDO.JK
BBCA.JK**

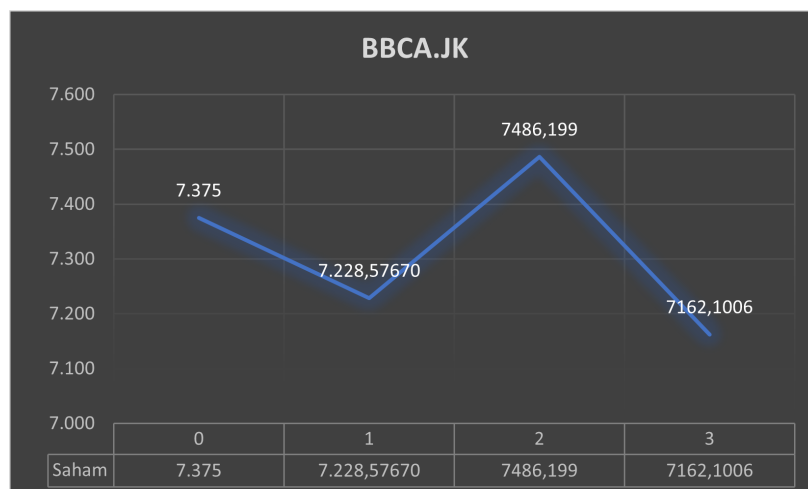
Hari	BBCA.JK	Status
0	7375.0	Belum punya saham dan beli
1	7228.5767	Tahan
2	7486.199	Jual (Untung = 214.6919)
3	7162.1006	

Tabel 2: Analisis Jual-Beli Saham BBKA.JK



(gambar di atas diambil dari program yang dijalankan pada 13/6/2022)

Gambar 10: Perbandingan Hasil Prediksi dan Harga Asli Saham BBKAJK



Gambar 11: Hasil Forecasting Saham BBKA.JK

UNVR.JK

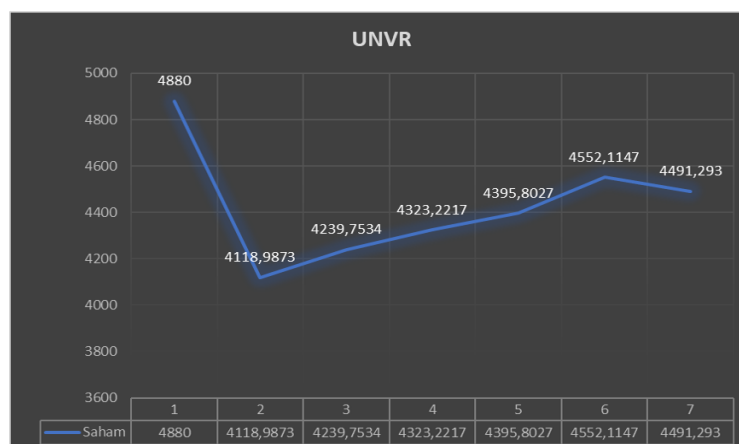
Hari	UNVR.JK	Status
0	4880.0	Belum punya saham dan tidak beli
1	4118.9873	Beli
2	4239.7534	Tahan
3	4323.2217	Tahan
4	4395.8027	Tahan
5	4552.1147	Jual (Untung = 407.54785)
6	4491.293	

Tabel 3: Analisis Jual-Beli Saham UNVR.JK



(gambar di atas diambil dari program yang dijalankan pada 13/6/2022)

Gambar 12: Perbandingan Hasil Prediksi dan Harga Asli Saham UNVR.JK



Gambar 13: Hasil Forecasting Saham UNVR.JK (dari program yang dijalankan pada 24/5/2022)

3.4 Analisis lainnya

3.4.1 *Trial and Error* dalam penentuan nilai Train:Test

Berikut perbandingan besar data test dan galat yang dihasilkannya. Dari tabel di bawah, kami menyimpulkan untuk memilih data test sebesar 4% untuk menghomogenkan program (penelitian lebih lanjut didapat bahwa, semakin besar data test, maka hasil running program semakin lebih lama).

Besar data test	SIDO.JK (error rata rata 10 hari)	BBCA.JK (error rata rata 10 hari)	UNVR.JK (error rata rata 10 hari)
1.5%	0.023199575	0.034479644	0.07344828
2%	0.02505995	0.032814685	0.07623534
3%	0.01516184	0.042827062	0.07232501
4%	0.019562311	0.036342464	0.06844522
5%	0.028493548	0.037816726	0.07659979
10%		0.039083783	
15%		0.03248696	
20%		0.036728024	

Tabel 4: Hasil 10 Hari ke Belakang Dimulai pada 24/5/2022 untuk Ketiga Saham

3.4.2 Up-down kebenaran forecasting aproksimasi vs data eksak

Metode *up-down* diperlukan dalam memberi keterangan lebih lanjut tentang seberapa akurat program melakukan prediksi dalam hal kejadian perubahan saham, apakah harga saham itu menaik, menurun, atau tidak kedua-duanya. Metode ini pada dasarnya membandingkan data historis suatu saham dengan hasil aproksimasi oleh program yang dijabarkan secara matematika sebagai berikut:

Jika $HSE_n < HSE_{n+1}$ dan $HSA_n < HSA_{n+1}$ atau $HSE_n > HSE_{n+1}$ dan $HSA_n > HSA_{n+1}$ atau $HSE_n = HSE_{n+1}$ dan $HSA_n = HSA_{n+1}$, maka kasus bernilai 1 (yang menandakan program berhasil memprediksi kejadian perubahan saham dengan benar).

Sebaliknya, jika salah satu premis diatas tidak terpenuhi, maka kasus bernilai 0 (menandakan program gagal memprediksi kejadian perubahan saham dengan benar). Tetapi pada kasus percobaan/kenyataan, LSTM **tidak mampu** memprediksi harga saham yang tidak mengalami perubahan atau $HS_n = HS_{n+1}$, sehingga untuk kasus ketika tidak terjadi perubahan/tidak naik atau turun, maka akan diberikan tanda khusus dengan “kasus ==”.

Sehingga untuk pengecekan 10 hari historis data harga saham didapat hasil sebagai berikut:

Hari n ke belakang	Kasus Saham SIDO.JK	Kasus Saham BBCA.JK	Kasus Saham UNVR.JK
0	1	0	1
1	kasus ==	kasus ==	1
2	0	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	1	1	0
7	kasus ==	1	1

8	0	1	1
9	kasus ==	1	1

Tabel 5: Up-down Forecasting Aproksimasi VS Data Eksak Berdasarkan 10 Hari Historis Data Harga Saham

Dimana perhitungan historis diambil pada 12/6/2022 dengan perhitungan mundur dari hari 12/6/2022 hingga 9 hari kebelakang (10 hari).

Dari hasil tabel, dapat disimpulkan bahwa rata-rata program mengaproksimasi perubahan **dengan benar** sebesar:

$$\frac{19}{30} = 0.633 = 63.33\%$$

3.4.3 Asumsi Besarnya Error pada Analisis Error Saham Unilever

Besar data test	SIDO.JK (error rata rata 10 hari)	BBCA.JK (error rata rata 10 hari)	UNVR.JK (error rata rata 10 hari)
1.5%	0.023199575	0.034479644	0.07344828
2%	0.02505995	0.032814685	0.07623534
3%	0.01516184	0.042827062	0.07232501
4%	0.019562311	0.036342464	0.06844522
5%	0.028493548	0.037816726	0.07659979
10%		0.039083783	
15%		0.03248696	
20%		0.036728024	

Tabel 4: Hasil 10 Hari ke Belakang Dimulai pada 24/5/2022 untuk Ketiga Saham

Pada tabel yang disajikan, kelompok kami mengambil kesimpulan bahwa besarnya error rata-rata yang dihasilkan oleh saham UNVR.JK **mungkin** dipengaruhi dampak internasional yang kala itu adalah perang Rusia dan Ukraina yang dapat dilihat secara pergerakan saham eksaknya sendiri (data harga saham Unilever 3 bulan kebelakang dari 17 Juni 2022):



Gambar 14: Saham Unilever (UNVR)

di mana Unilever mengalami kemerosotan harga saham. Hal ini menyebabkan metode LSTM sulit dalam memprediksi harga saham dikarenakan sangat fluktuatifnya harga saham Unilever itu sendiri (Asumsi kelompok kami belum tentu benar, dikarenakan kurangnya data yang mendukung untuk mempertegas kebenaran dari asumsi besarnya error pada prediksi harga saham LSTM).

BAB IV

PENUTUPAN

4.1 Kesimpulan

Berdasarkan rumusan masalah yang ada pada bagian pendahuluan, bisa dibuat kesimpulan sebagai berikut secara berurutan sesuai dengan rumusan masalah.

1. Saham adalah suatu alat bukti atas kepemilikan dari sebuah perusahaan yang berupa lembaran-lembaran surat berharga. Adanya perkembangan teknologi memungkinkan robot akan mencari peluang *open trade*, *open sell*, dan *buy* di pasar saham yang kemudian 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 strategi *trading*, yang mana itu di luar cakupan proyek ini.

Terdapat dua metode yang digunakan oleh *stock trading bot*. Yang pertama adalah *AutoRegressive Integrated Moving Average* (ARIMA) yang memanfaatkan persamaan linier tipe regresi dan yang kedua adalah *Long Short Term Memory* (LSTM) yang memanfaatkan komponen kunci yang disebut *cell state* (Ct) yang mewakili ingatan jangka panjang dan jangka pendek internal sel.

Berdasarkan ketiga data historis harga saham penutupan yang digunakan, dapat dilihat bahwa baik untuk BBKA.JK, UNVR.JK dan SIDO, LSTM lebih mendekati harga saham asli dibandingkan dengan ARIMA. Dari metrik evaluasi sendiri, LSTM memiliki tingkat RMSE lebih baik daripada ARIMA, sehingga dapat disimpulkan bahwa LSTM adalah metode yang cocok untuk pendekatan robot trading dalam memprediksi harga saham.

2. Untuk memvisualisasikan hasil forecasting harga saham dengan metode LSTM, bisa dilakukan dengan cara yang terdapat di bagian 3.2 dan 3.3. Output hasil visualisasi bisa terlihat pada gambar 8-13 dan tabel 1-3 di bagian 3.3.2.
3. Untuk kriteria jual beli atau beli saham, rincian jelasnya dapat dilihat di bagian 3.2.3. Namun, secara singkat robot merekomendasikan menjual saham saat harga saham menurun dan jika harga jual setelah ongkos jual lebih besar dari harga beli setelah ongkos beli. Robot akan merekomendasikan untuk tidak membeli ataupun menjual (tahan) saham saat harga saham hari ini lebih besar atau sama dengan harga saham kemarin, atau saat harga jual setelah ongkos jual lebih kecil dari harga beli setelah

ongkos beli. Ada biaya yang dikenakan dalam tiap transaksi ini, yang dapat mempengaruhi keputusan jual beli saham. Transaksi saham ini dikenakan ongkos yang bervariasi. Ongkos beli sebesar 0.19% ditambahkan ke harga pembelian, sedangkan ongkos jual yaitu 0.39% diambil dari hasil penjualan saham.

4. Kesimpulan lainnya yang kelompok kami dapat adalah:
 - Untuk bagian *up-down* dari program LSTM, didapati bahwa LSTM **tidak mampu** memprediksi harga saham yang tidak mengalami perubahan atau $HS_n = HS_{n+1}$, sehingga untuk kasus ketika tidak terjadi perubahan/tidak naik atau turun, maka akan diberikan tanda khusus dengan “kasus ==”.
 - Kasus Unilever memiliki tingkat error rata-rata yang paling besar dikarenakan dinamisnya harga saham yang terjadi pada bulan Mei-Juni.

4.2 Saran

Adapun saran dari kelompok kami antara lain:

- Sekiranya dibuat semacam rencana kontingensi otomatis jika prediksi salah
- Robot trading tidak disarankan untuk kegiatan trading tanpa supervisi karena kurangnya ketelitian dari robot trading dalam memprediksi dikarenakan keterbatasan perangkat keras/lunak yang kami gunakan
- Dapat memungkinkan dikembangkan metode LSTM baru dengan performa lebih baik dalam prediksi pergerakan saham
- Pengguna juga disarankan memperluas pengetahuan dari pendapat ahli bagaimana menanggapi kasus jual/beli/tahan dari robot trading.

DAFTAR PUSTAKA

Yahoo. Yahoo!Finance. <https://finance.yahoo.com/>.

Badan Penerbit Fakultas Hukum Universitas Indonesia, Jakarta. Rahadrjo, Sapto, 2004, Panduan

Shekhar, Shraddha. 2021. "LSTM for Text Classification in Python". <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>. Diakses pada 20 Mei 2022.

Baheti, Pragati. 2022. "12 Types of Neural Network Activation Functions: How to Choose?". [https://www.v7labs.com/blog/neural-networks-activation-functions/#:~:text=Tanh%20Function%20\(Hyperbolic%20Tangent\)&text=In%20Tanh%2C%20the%20larger%20the,output%20will%20be%20to%20%2D1.0](https://www.v7labs.com/blog/neural-networks-activation-functions/#:~:text=Tanh%20Function%20(Hyperbolic%20Tangent)&text=In%20Tanh%2C%20the%20larger%20the,output%20will%20be%20to%20%2D1.0). Diakses pada 20 Mei 2022.

Saeed, Mehreen. 2021. "A Gentle Introduction to Sigmoid Function". <https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/#:~:text=A%20sigmoid%20unit%20in%20a,the%20weighted%20sum%20of%20inputs>. Diakses pada 21 Mei 2022

PT Bursa Efek Indonesia. (2021). PT Bursa Efek Indonesia. <https://www.idx.co.id/investhub/perpajakan/>. Diakses pada 23 Mei 2022

Indonesia. Undang-Undang Nomor 36 Tahun 2008 tentang Perubahan Keempat atas Undang-undang Nomor 7 Tahun 1983 tentang Pajak Penghasilan. Sekretariat Negara. Jakarta.

Shekhar, Shraddha. 2021. "LSTM for Text Classification in Python". <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>. Diakses pada 11-4-2022

(2021, August). Machinelearningplus.com. <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>

Baheti, Pragati. 2022. "12 Types of Neural Network Activation Functions: How to Choose?". [https://www.v7labs.com/blog/neural-networks-activation-functions/#:~:text=Tanh%20Function%20\(Hyperbolic%20Tangent\)&text=In%20Tanh%2C%20the%20larger%20the,output%20will%20be%20to%20%2D1.0](https://www.v7labs.com/blog/neural-networks-activation-functions/#:~:text=Tanh%20Function%20(Hyperbolic%20Tangent)&text=In%20Tanh%2C%20the%20larger%20the,output%20will%20be%20to%20%2D1.0). Diakses pada 11-4-2022

Saeed, Mehreen. 2021. "A Gentle Introduction to Sigmoid Function".
<https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/#:~:text=A%20sigmoid%20unit%20in%20a,the%20weighted%20sum%20of%20inputs>. Diakses pada 11-4-2022

How to Build ARIMA Model in Python for time series forecasting? (2022). ProjectPro.
<https://www.projectpro.io/article/how-to-build-arima-model-in-python/544>

Agung. (2017, March 21). *ARIMA SARIMA : Si Kembar dari Time Series*. Agung Budi Santoso | Membumikan Ilmu Pengetahuan; Agung Budi Santoso.
<https://agungbudisantoso.com/arima-sarima-si-kembar-dari-time-series/>

Maklin, C. (2019, May 25). *ARIMA Model Python Example — Time Series Forecasting*. Medium; Towards Data Science.
<https://towardsdatascience.com/machine-learning-part-19-time-series-and-autoregressive-integrated-moving-average-model-arima-c1005347b0d7>

(2021, August). Machinelearningplus.com.
<https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>

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

Google Colab. Pemodelan Robot Trading.
<https://colab.research.google.com/drive/1cf7dO212NKyDAmM-VedCBS44if2BMLKT#scrollTo=uo2Tw2PYEYV9>

Google Colab. Pemodelan Robot Trading (sederhana).
<https://colab.research.google.com/drive/1qGiJaZ0j2kGaQIdKjFekPRVueMlh09YQ#scrollTo=c-PR2gQIEVJZ>

LAMPIRAN

Berikut adalah lampiran program robot trading yang kami buat pada *google collabs*:

```
[ ] 1 pip install yfinance --upgrade --no-cache-dir
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting yfinance
  Downloading yfinance-0.1.70-py2.py3-none-any.whl (26 kB)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (from yfinance) (1.21.6)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.7/dist-packages (from yfinance) (0.0.10)
Collecting lxml>=4.5.1
  Downloading lxml-4.9.0-cp37m-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_24_x86_64.whl (6.4 MB)
    |████████████████████████████████████████| 6.4 MB 7.2 MB/s
Requirement already satisfied: pandas>=0.24.0 in /usr/local/lib/python3.7/dist-packages (from yfinance) (1.3.5)
Collecting requests>=2.26
  Downloading requests-2.28.0-py3-none-any.whl (62 kB)
    |████████████████████████████████████████| 62 kB 31.5 MB/s
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.0->yfinance) (2022.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.0->yfinance) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>=0.24.0->yfinance) (1.15.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.26->yfinance) (1.24.3)
Requirement already satisfied: charset-normalizer<=2.0.0 in /usr/local/lib/python3.7/dist-packages (from requests>=2.26->yfinance) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.26->yfinance) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.26->yfinance) (2022.5.18.1)
Installing collected packages: requests, lxml, yfinance
Attempting to uninstall: requests
```

```
[ ] 1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 import yfinance as yf
5 import datetime as dt
6 from datetime import datetime
7 from sklearn.preprocessing import MinMaxScaler
8 from tensorflow.keras.models import Sequential
9 from tensorflow.keras.layers import Dense, Dropout, LSTM
10 from sklearn.metrics import mean_squared_error
11
12
13
14 #untuk load data saham (ticker)
15 company=str(input("Masukan Ticker saham yang ingin dianalisis program, sebagai contoh (BBCA.JK): "))# 'UNWR.JK'#atau SDO.JK; BBKA.JK
16
17 #Penentuan penanggalan mulai dan berakhir
18 start=dt.datetime(2012,1,1)
19 end=dt.date.today()
20
21 data = yf.download(company, start,end)
22 data = data.dropna()
23 data_close=data["Close"]
24
```

```
21 data = yf.download(company, start,end)
22 data = data.dropna()
23 data_close=data["Close"]
24
25 error_toleransi=0.04#0.031983333
26
27 status_punya_saham=str(input("Apakah anda sedang 'tidak' memiliki saham atau 'ingin membeli' saham baru? Jawab dengan ya/tidak: "))
28 if status_punya_saham=="ya":
29     punya_saham=0
30 else:
31     punya_saham=1
32
33 kasus_pajak_beli=str(input("Apakah anda menginginkan 'ongkos beli' (pajak beli) berbeda (bawaan adalah 0.0019), jawab dengan ya/tidak: "))
34 if kasus_pajak_beli=="ya":
35     pajak_beli=float(input("Masukan besarnya 'ongkos beli' (pajak beli) baru dengan decimal: "))
36 else:
37     pajak_beli=0.0019
38
39 kasus_pajak_jual=str(input("Apakah anda menginginkan 'ongkos jual' (pajak jual) berbeda (bawaan adalah 0.0039), jawab dengan ya/tidak: "))
40 if kasus_pajak_jual=="ya":
41     pajak_jual=float(input("Masukan besarnya 'ongkos jual' (pajak jual) baru dengan decimal: "))
42 else:
43     pajak_jual=0.0039
```

```

39 kasus_pajak_jual=str(input("Apakah anda menginginkan 'ongkos jual' (pajak jual) berbeda (bawaan adalah 0.0039), jawab dengan ya/tidak: "))
40 if kasus_pajak_jual=="ya":
41     pajak_jual=float(input("Masukan besarnya 'ongkos jual' (pajak jual) baru dengan decimal: "))
42 else:
43     pajak_jual=0.0039
44
45 list_banyak_error_toleransi=[0]
46
47 lompatan=1
48
49
50 harga_beli_saham=data_close.iloc[len(data_close.index)-1] + pajak_beli * data_close.iloc[len(data_close.index)-1]
51
52
53 while lompatan<11:
54     end=dt.date.today()+dt.timedelta(days=lompatan)#Days bisa diganti bersesuaian
55
56     #mempersiapkan datanya
57     scaler=MinMaxScaler(feature_range=(0,1)) #dia normalisasi data
58     scaled_data=scaler.fit_transform(data_close.values.reshape(-1,1))
59

```

```

56 #mempersiapkan datanya
57 scaler=MinMaxScaler(feature_range=(0,1)) #dia normalisasi data
58 scaled_data=scaler.fit_transform(data_close.values.reshape(-1,1))
59
60 #Prediksi forecasting berdasarkan 60 hari sebelumnya????
61 prediction_days=round(4/100*data_close.size) #ini banyak hari untuk testing data
62 #ini tadinya 60 hari
63
64
65 #list untuk hasil training
66 x_train=[]#kayaknya ini
67 y_train=[]
68
69 #kasus ini kenapa tidak pengambilan acak, karena kita mau belajar dari historisnya, buat prediksi si 60 hari itu nanti
70 for x in range(prediction_days, len(scaled_data)):#ini x dari 120 sampai akhir ujung hasil minmax data close (2000an)
71     x_train.append(scaled_data[x-prediction_days:x, 0])
72     y_train.append(scaled_data[x, 0])
73
74 x_train, y_train= np.array(x_train), np.array(y_train)
75 x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1],1))

```

```

74 x_train, y_train= np.array(x_train), np.array(y_train)
75 x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1],1))
76
77
78 #membuat model
79 model=Sequential()
80
81 model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1], 1)))
82 model.add(Dropout(0.2))
83
84 model.add(LSTM(units=50, return_sequences=True))
85 model.add(Dropout(0.2))
86
87 model.add(LSTM(units=50))
88 model.add(Dropout(0.2))
89
90 model.add(Dense(units=1))#prediction untuk next closenya
91
92 #=====
93 model.compile(optimizer="adam", loss= "mean_squared_error")
94 model.fit(x_train, y_train, epochs=1, batch_size=32)#nanti epochs balikin ke 25
95

```

```

92 #=====
93 model.compile(optimizer="adam", loss= "mean_squared_error")
94 model.fit(x_train, y_train, epochs=1, batch_size=32)#nanti epochs balikin ke 25
95
96
97 #test model akurasi dari data yang dipunya
98 #load test datanya
99 #===== ada catatan di bawah:
100 test_data=data_close#yf.download(company, start,end)
101 test_data=test_data.dropna()
102 actual_prices=data_close.values
103
104 #bagian concatnya keknya rada-rada...
105 total_dataset=pd.concat((data_close, test_data), axis=0)
106
107 #ada unsur kesengajaan di total_dataset, concatnya 2 hal yang sama...
108 #beigtu juga pada model_inputs
109 #dibagian total_dataset, langsung aja pake len(test_data)
110
111
112 model_inputs = total_dataset[len(total_dataset) - len(test_data) - prediction_days:].values#yang dipake buat belajar?
113 model_inputs = model_inputs.reshape(-1,1)

```

```

109 #dibagian total_dataset, langsung aja pake len(test_data)
110
111
112 model_inputs = total_dataset[len(total_dataset) - len(test_data) - prediction_days:].values#yang dipake buat belajar?
113 model_inputs = model_inputs.reshape(-1,1)
114 model_inputs = scaler.transform(model_inputs)
115 #=====
116
117 #hasil dari prediction dari test data
118 x_test = []
119
120 for x in range(prediction_days, len(model_inputs)):
121     x_test.append(model_inputs[x-prediction_days:x,0])
122
123 x_test = np.array(x_test)
124 x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
125
126 predicted_prices = model.predict(x_test)
127 predicted_prices = scaler.inverse_transform(predicted_prices)
128
129 rmse=np.sqrt(mean_squared_error(actual_prices,predicted_prices))
130 #Memprediksi saham sekian hari kedepan
131 real_data = [model_inputs[len(model_inputs) - prediction_days:len(model_inputs)+1], 0]]
132 real_data = np.array(real_data)
133 real_data = np.reshape(real_data, (real_data.shape[0], real_data.shape[1], 1))
134

```

```

130 #Memprediksi saham sekian hari kedepan
131 real_data = [model_inputs[len(model_inputs) - prediction_days:len(model_inputs)+1], 0]]
132 real_data = np.array(real_data)
133 real_data = np.reshape(real_data, (real_data.shape[0], real_data.shape[1], 1))
134
135 prediction = model.predict(real_data)
136 prediction = scaler.inverse_transform(prediction)
137 print("Maka hasil forecasting saham untuk ke\t",dt.date.today()+dt.timedelta(days=lompatan),\
138       "\tadalah:", "\n" f"ini adalah hasil prediksinya {prediction}",\
139       "\nDimana RMSEnya adalah:", rmse, "\n")
140
141 data_close.loc[len(data_close.index)] = prediction #append hasil predict
142
143 print(data_close.iloc[len(data_close.index)-2], f"harga saat hari ke: {lompatan-1}\n")
144 print(data_close.iloc[len(data_close.index)-1], f"harga saat hari ke: {lompatan}\n")
145
146 keadaan_saham_error=abs(data_close.iloc[len(data_close.index)-2] - data_close.iloc[len(data_close.index)-1])/data_close.iloc[len(data_close.index)-2]
147 #harga_beli_saham=data_close.iloc[len(data_close.index)-1] + pajak_beli * data_close.iloc[len(data_close.index)-1]
148
149 if punya_saham==0:
150     if data_close.iloc[len(data_close.index)-2] < data_close.iloc[len(data_close.index)-1]:
151         punya_saham=1
152         harga_beli_saham=data_close.iloc[len(data_close.index)-2] + pajak_beli * data_close.iloc[len(data_close.index)-2]
153         print(f"Beli saham pada hari ke {lompatan-1}, karena besok akan naik\n")
154         print(f"Harga saham yang dibeli adalah {data_close.iloc[len(data_close.index)-2]}\n")
155

```

```

155
156 if punya_saham==1:
157     if lompatan==1:
158         if data_close.iloc[len(data_close.index)-2] > data_close.iloc[len(data_close.index)-1]:
159             #if keadaan_saham_error > error_toleransi:
160                 harga_jual_saham = data_close.iloc[len(data_close.index)-1] - pajak_jual * data_close.iloc[len(data_close.index)-1]
161                 if harga_beli_saham < harga_jual_saham:
162                     print("Saham harus dijual saat hari: ", lompatan-1, "karena besok akan jatuh \n")
163                     print("Keuntungan yang akan didapat persatuan sahamnya sebesar: ", harga_jual_saham - harga_beli_saham)
164                     lompatan=lompatan+100
165
166             else:
167                 print("Saham merugi, tetapi saham tidak bisa dijual saat hari: ", lompatan-1, ""karena tidak didapat keuntungan dari
168                     selisih harga jual saham dan harga beli saham setelah pajaknya
169                     dengan nilai: \n"", f"ini adalah nilai harga jual saham setelah pajak: {harga_jual_saham}\n",\
170                     f"ini adalah nilai harga beli sahamnya setelah pajak: {harga_beli_saham}")
171
172         else:
173             #list_banyak_error_toleransi=[0]
174             print("Saham dalam keadaan baik dan meningkat untuk hari prediksi ke: ", lompatan, "\n")
175

```

```

176
177 else:
178     if data_close.iloc[len(data_close.index)-2] > data_close.iloc[len(data_close.index)-1]:
179         #if keadaan_saham_error > error_toleransi:
180             harga_jual_saham=data_close.iloc[len(data_close.index)-2] - pajak_jual * data_close.iloc[len(data_close.index)-2]
181             if harga_beli_saham < harga_jual_saham:
182                 print("Saham harus dijual saat hari: ", lompatan-1, "karena besok akan jatuh \n")
183                 print("Keuntungan yang akan didapat persatuan sahamnya sebesar: ", harga_jual_saham - harga_beli_saham)
184                 lompatan=lompatan+100
185
186             else:
187                 print("Saham merugi, tetapi saham tidak bisa dijual saat hari: ", lompatan-1, ""karena tidak didapat keuntungan dari
188                     selisih harga jual saham dan harga beli saham setelah pajaknya
189                     dengan nilai: \n"", f"ini adalah nilai harga jual saham setelah pajak: {harga_jual_saham}\n",\
190                     f"ini adalah nilai harga beli sahamnya setelah pajak: {harga_beli_saham}")
191
192
193             #else:
194                 #print("Keadaan sedang merugi, tetapi masih dibawah batas toleransi pada hari prediksi ke: ", lompatan, "\n")
195                 #list_banyak_error_toleransi.append(1)
196
197         else:
198             #list_banyak_error_toleransi=[0]

```

```

199
200 #if sum(list_banyak_error_toleransi)>=2:
201     #print("kejadian saham merugi sudah 2 kali, maka program stop dan disarankan untuk menjual pada hari ke: ",lompatan-2)
202     #lompatan=100
203
204
205 lompatan=lompatan+1
206
207 #Plotting test prediksinya dan eksaknya
208 plt.figure(figsize=(15,7))
209 plt.grid(True)
210 plt.plot(actual_prices[:len(actual_prices)-(lompatan)], color="green", label=f"Actual {company} Price")
211 plt.plot(predicted_prices, color="pink", label=f"Predicted {company} Price")
212 plt.title(f"{company} Share Price")
213 plt.xlabel("Time")
214 plt.ylabel(f"{company} Share Price")
215 plt.legend()
216 plt.show()
217

```

Dengan Output Program:

```
➤ Masukan Ticker saham yang ingin dianalisis program, sebagai contoh (BBCA.JK): SIDO.JK
[*****100%*****] 1 of 1 completed
Apakah anda sedang 'tidak' memiliki saham atau 'ingin membeli' saham baru? Jawab dengan ya/tidak: ya
Apakah anda menginginkan 'ongkos beli' (pajak beli) berbeda (bawaan adalah 0.0019), jawab dengan ya/tidak: tidak
Apakah anda menginginkan 'ongkos jual' (pajak jual) berbeda (bawaan adalah 0.0039), jawab dengan ya/tidak: tidak
63/63 [=====] - 32s 243ms/step - loss: 0.0113
Maka hasil forecasting saham untuk ke 2022-06-14 adalah:
ini adalah hasil prediksinya [[871.3399]]
Dimana RMSEnya adalah: 56.462198302773196

975.0 harga saat hari ke: 0

[[871.3399]] harga saat hari ke: 1

63/63 [=====] - 30s 245ms/step - loss: 0.0100
Maka hasil forecasting saham untuk ke 2022-06-15 adalah:
ini adalah hasil prediksinya [[926.7401]]
Dimana RMSEnya adalah: 46.34020348117902

[[871.3399]] harga saat hari ke: 1

[[926.7401]] harga saat hari ke: 2

[[871.3399]] harga saat hari ke: 1

[[926.7401]] harga saat hari ke: 2

Beli saham pada hari ke 1, karena besok akan naik

Harga saham yang dibeli adalah [[871.3399]]

Saham dalam keadaan baik dan meningkat untuk hari prediksi ke: 2

63/63 [=====] - 25s 196ms/step - loss: 0.0108
Maka hasil forecasting saham untuk ke 2022-06-16 adalah:
ini adalah hasil prediksinya [[935.2284]]
Dimana RMSEnya adalah: 44.284198458363115

[[926.7401]] harga saat hari ke: 2

[[935.2284]] harga saat hari ke: 3

Saham dalam keadaan baik dan meningkat untuk hari prediksi ke: 3
```

Saham dalam keadaan baik dan meningkat untuk hari prediksi ke: 3

64/64 [=====] - 31s 228ms/step - loss: 0.0117
Maka hasil forecasting saham untuk ke 2022-06-17 adalah:
ini adalah hasil prediksinya [[950.30054]]
Dimana RMSEnya adalah: 45.152802031603876

[[935.2284]] harga saat hari ke: 3

[[950.30054]] harga saat hari ke: 4

Saham dalam keadaan baik dan meningkat untuk hari prediksi ke: 4

64/64 [=====] - 31s 244ms/step - loss: 0.0114
Maka hasil forecasting saham untuk ke 2022-06-18 adalah:
ini adalah hasil prediksinya [[937.38495]]
Dimana RMSEnya adalah: 44.71886310428583

[[950.30054]] harga saat hari ke: 4

[[937.38495]] harga saat hari ke: 5



[[950.30054]] harga saat hari ke: 4

[[937.38495]] harga saat hari ke: 5

Saham harus dijual saat hari: 4 karena besok akan jatuh

Keuntungan yang akan didapat persatuan sahamnya sebesar: [[73.59894]]

SIDO JK Share Price



Saham harus dijual saat hari: 4 karena besok akan jatuh

Keuntungan yang akan didapat persatuan sahamnya sebesar: [[73.59894]]

