

# RANCANG BANGUN APLIKASI PREDIKSI HARGA SAHAM MENGUNAKAN *DEEP LEARNING* DENGAN ALGORITMA *LONG SHORT-TERM MEMORY*

Gesang Paudra Jaya

Gesang Paudra Jaya, Sistem Informasi, STMIK Bani Saleh, g.paudra@gmail.com

## Abstrak

Saham adalah suatu instrumen keuangan yang sulit untuk diprediksi, sebelumnya cara prediksi harga saham dilakukan dengan menganalisis data-data perdagangan sebelumnya secara manual, namun ini hanya berguna untuk menentukan apakah saham akan bergerak naik atau turun dengan tingkat akurasi yang tidak bisa diukur secara pasti. Oleh karena itu dibangunlah sebuah aplikasi yang menggunakan deep learning dengan algoritma long short-term memory (LSTM). Aplikasi ini menggunakan data perdagangan saham berupa harga open, close, high, low, volume perdagangan, dan kode saham agar dapat memprediksi harga closing kedepannya. Data yang digunakan pada aplikasi ini menggunakan data dari saham perusahaan Microsoft dan saham perusahaan Bank Central Asia (BCA), aplikasi telah diujicoba dan berhasil memprediksi harga saham dengan tingkat rata-rata akurasi keseluruhan mencapai 97,4% yang didapatkan dengan menghitung nilai rata-rata dari seluruh hasil akurasi prediksi.

**Kata Kunci:** *Deep Learning*, Saham, Prediksi harga saham, *Long Short-Term Memory*.

## PENDAHULUAN

Saham adalah tanda penyertaan modal seseorang atau pihak (badan usaha) dalam suatu perusahaan atau perseroan terbatas. Dengan menyertakan modal tersebut, maka pihak tersebut memiliki klaim atas pendapatan perusahaan, klaim atas aset perusahaan, dan berhak hadir dalam Rapat Umum Pemegang Saham (RUPS) (Bursa Efek Indonesia (BEI), 2010). Naik dan turunnya nilai atau harga saham tergantung dari performa keuangan perusahaan, selain itu, keadaan sosial dan ekonomi juga bisa menjadi sentimen atau pengaruh terhadap pergerakan harga saham, maka dari itu harus dilakukan sebuah prediksi untuk menentukan apakah dalam beberapa waktu kedepan saham akan naik atau turun. Prediksi saham sendiri adalah suatu proses yang dilakukan secara sistematis untuk menentukan nilai-nilai yang paling mungkin terjadi pada saham di waktu yang akan datang berdasarkan informasi masa lalu dan informasi yang dimiliki saat ini.

Pada awalnya untuk memprediksi harga saham, para investor melihat dan mencatat data historis saham dan menentukan pola-pola pada data tersebut secara manual, namun cara ini sangat tidak praktis dikarenakan para investor secara manual harus mencatat dan memperhitungkan segala hal untuk

menentukan apakah saham akan naik atau turun dalam beberapa waktu kedepan, dan itu hanya 1 saham, banyak investor yang mereka berinvestasi pada banyak saham, yang membuat tingkat kesulitan prediksi secara manual tersebut meningkat berkali lipat.

Dari permasalahan tersebut maka terbentuklah penelitian ini, yaitu sebuah program atau aplikasi untuk membantu para investor dalam mencari perusahaan yang memiliki kinerja saham dan keuangan yang baik, yaitu dimana pengguna bisa melihat apakah dalam waktu yang ditentukan harga saham perusahaan naik atau turun. Program ini peneliti buat dengan menggunakan pendekatan yang dinamakan *Deep Learning*. *Deep Learning* adalah suatu teknik pengembangan dari ML (*Machine Learning*) yang dimana sebuah program komputer membangun suatu jaringan atau *network* yang mirip dengan jaringan *neuron* pada otak manusia yang membuat proses pengembangan dan prediksi dari program lebih cepat dan lebih akurat dari sebelumnya.

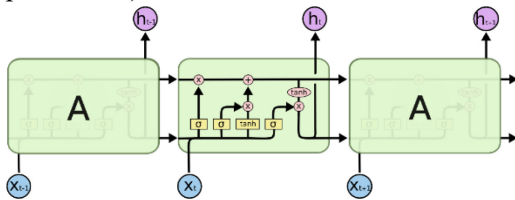
## METODE

### A. Deep learning

*Deep learning* adalah bagian dari pembelajaran mesin atau Machine Learning, yang pada dasarnya adalah neural network atau jaringan saraf dengan tiga atau lebih lapisan. Jaringan saraf ini mencoba untuk mensimulasikan perilaku otak manusia meskipun jauh dari kemampuan yang memungkinkannya untuk belajar dari sejumlah besar data. Sementara jaringan saraf dengan satu lapisan masih dapat membuat perkiraan perkiraan, lapisan tersembunyi tambahan dapat membantu mengoptimalkan dan menyempurnakan akurasi (IBM, 2020).

### B. Long Short-Term Memory (LSTM)

LSTM adalah sebuah jaringan saraf yang memiliki *feedback connection* yang membuatnya berbeda dengan recurrent neural network. Properti ini memungkinkan LSTM untuk memproses seluruh urutan data (misalnya deret waktu) tanpa memperlakukan setiap titik dalam urutan secara independen, melainkan, mempertahankan informasi yang berguna tentang data sebelumnya dalam urutan untuk membantu pemrosesan titik data baru. Akibatnya, LSTM sangat baik dalam memproses urutan data seperti teks, ucapan, dan deret waktu (Dolphin, 2020).



Gambar 1. Diagram LSTM

Sumber : Colah, 2015

LSTM menggunakan beberapa *gate* untuk mengontrol bagaimana informasi didalam data masuk, disimpan, dan keluar dari jaringan, ketiga *gate* tersebut adalah *input gate*, *output gate*, *forget gate*. Ketiga gerbang ini juga dianggap sebagai *filter* dan masing-masing memiliki neural network sendiri.

#### 1. Forget Gate

*Forget gate* atau gerbang lupa adalah gate yang menentukan apakah data *input*  $X_t$  dan *output*  $h_{(t-1)}$  bisa lewat atau tidak. Keputusan ini dibuat oleh fungsi sigmoid yang menghasilkan nilai  $f_t$ . Pada output ini, nilai mendekati 1 berarti data “boleh lanjut” atau lewat, sedangkan mendekati 0 berarti data “tidak boleh lewat” atau “abaikan”.

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

#### 2. Input Gate

Input Gate adalah gate yang menentukan bagian mana pada jaringan yang akan diperbarui atau *update*, input gate memiliki dua bagian *layer*, yaitu *layer* dengan fungsi sigmoid dan *layer* dengan fungsi tanh. *Layer* dengan fungsi sigmoid menentukan nilai mana yang diperbarui dan menghasilkan nilai  $i_t$  kemudian *layer* dengan fungsi tanh memberikan *weight* atau bobot pada nilai-nilai yang dilewati, kemudian menentukan tingkat kepentingan dan menghasilkan nilai  $\tilde{C}_t$ . Kemudian kedua fungsi ini bisa digabung dengan cara mengkalikan kedua nilai, selanjutnya adalah mengkalikan  $C_{(t-1)}$  kondisi sebelumnya dengan nilai *forget gate*  $f_t$ . Setelah itu menambahkan dengan hasil perkalian  $i_t$  dan  $\tilde{C}_t$  sehingga menciptakan *cell state*  $C_t$  (Aprian, 2020).

$$i_t = \sigma(U_i X_t + W_i h_{t-1} + b_i) \quad (2)$$

#### 3. Cell State

Selanjutnya memperbarui nilai cell state lama  $C_{t-1}$  menjadi cell state baru  $\tilde{C}_t$  dan  $C_t$

$$\tilde{C}_t = \tanh(U_c X_t + W_c h_{t-1} + b_c) \quad (3)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (4)$$

#### 4. Output Gate

Terakhir adalah *output gate*, *gate* ini memutuskan apa yang akan dihasilkan, pertama menjalankan fungsi sigmoid untuk menentukan *cell* apa yang akan dihasilkan pada  $O_t$ . Selanjutnya mengkalikan dengan *cell state*  $C_t$  yang sudah melewati fungsi tanh dan hasilnya menjadi  $h_t$  kemudian meneruskannya ke *cell* selanjutnya yaitu  $h_{(t-1)}$ . *Output gate* ini tidak berkontribusi untuk *cell state*, namun *gate* inilah yang membedakan *cell state* dan *output*  $h_t$  yang sebenarnya (Aprian, 2020)

$$O_t = \sigma(U_o X_t + W_o h_{t-1} + b_o) \quad (5)$$

$$h_t = O_t * \tanh(C_t) \quad (6)$$

Selanjutnya untuk menentukan akurasi dari prediksi yang dihasilkan, maka peneliti menggunakan formula akurasi sebagai berikut :

$$A = 1 - \left| \frac{N_p}{N_a - 1} \right| \times 100\% \quad (7)$$

$A$  = Nilai akurasi dalam pesen

$N_p$  = Nilai Prediksi

$N_a$  = Nilai Sebenarnya

Kemudian untuk menentukan rata-rata keseluruhan akurasi prediksi, maka peneliti menggunakan formula sebagai berikut :

$$R_t = \frac{\bar{x}}{n} \times 100\% \quad (8)$$

$R_t$  = Nilai rata – rata keseluruhan

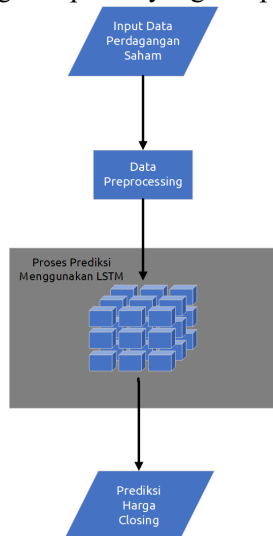
$\bar{x}$  = Jumlah total akurasi prediksi

$n$  = Banyaknya data akurasi prediksi

## HASIL DAN PEMBAHASAN

### A. Implementasi LSTM

Sebelum program ini dibangun, terlebih dahulu dibuat sebuah desain dari program yang akan dibangun. Desain ini akan menjadi gambaran besar seperti apa program yang akan dibangun nantinya. Dari desain ini kita juga akan melihat peranan masing-masing komponen yang ada pada program.

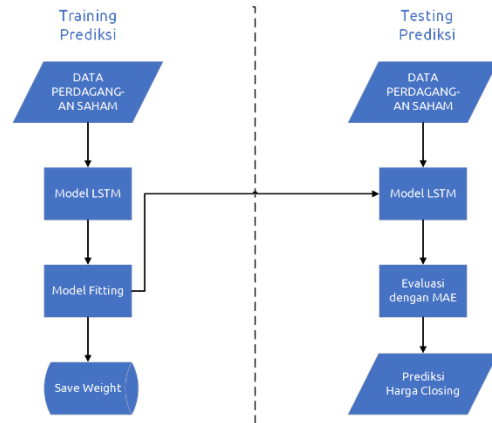


Gambar 2. Desain alur program secara sederhana.

Secara sederhana, program yang dibangun nanti terdiri dari 5 bagian utama, yaitu :

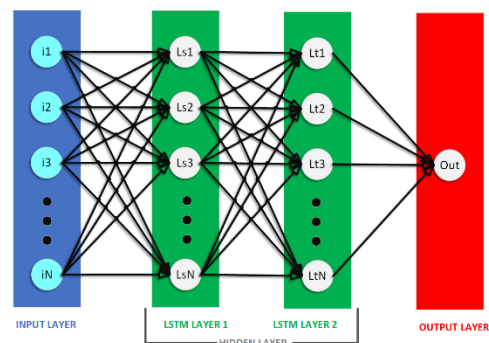
- Input data
- Data preprocessing.
- Membuat Model LSTM
- Training Model LSTM
- Output dan Evaluasi hasil prediksi

Sebelum LSTM diimplementasikan untuk memprediksi harga saham, perlu dibuatnya model jaringan saraf LSTM dan melatih atau training jaringan saraf LSTM ini sehingga model ini dapat mengenali dan paham akan tugasnya yaitu memprediksi harga closing saham. Pada tahap latihan atau training, terdapat 2 model, yaitu model training dan model testing.



Gambar 3. Skema Latihan Prediksi Harga Saham

Dalam program prediksi harga saham ini, untuk mendapatkan akurasi prediksi yang baik, maka LSTM perlu dilatih dengan sejumlah data *training*. Tujuan dari melatih LSTM adalah untuk menemukan ciri dari setiap data yang diberikan kemudian menandai neuron-neuron yang ada pada neural network atau jaringan saraf, oleh karena itu perlu dibuatnya sebuah model untuk melakukan latihan agar saat nanti dilakukan pengujian untuk melakukan prediksi, maka program sudah terlatih dengan baik. Sebelum memulai latihan, perlu dibuatnya sebuah model LSTM, disini peneliti sudah membuat visualisasi bagaimana model LSTM tersusun.



Gambar 4. Visualisasi *layers* yang digunakan

Pada gambar diatas, dapat dilihat bahwa model LSTM yang peneliti bangun memiliki 4 layers atau lapisan, mulai dari kiri yaitu terdapat lapisan pertama yang disebut dengan input layer, pada lapisan ini data dari dataset yang sudah dibentuk dimasukkan, dapat dilihat bahwa lapisan ini terdiri dari cell i1 sampai dengan cell iN, jumlah cell ini tergantung dari berapa banyak data perdagangan saham yang ditentukan oleh admin, misalnya disini peneliti memasukkan nilai 80, berarti program akan menggunakan 80 hari terakhir data perdagangan sebagai dataset yang digunakan untuk melatih model prediksi.

Setelah mendapatkan hasil prediksi, tahap berikutnya adalah menampilkan hasil prediksi pada *client-side*, disini peneliti menggunakan Anvil Works sebagai *client-side* untuk aplikasi prediksi harga saham ini, penggunaan *client-side* ini bertujuan untuk memudahkan para pengguna yang ingin mencoba aplikasi ini dengan mudah, pengguna hanya cukup memasukkan kode saham dan menekan satu tombol maka akan muncul informasi hasil prediksi yang didapatkan dari aplikasi prediksi harga saham ini.

Proses training yang telah dilakukan menggunakan 1000 epochs pada data saham BBCE dan 1000 epochs pada data saham MSFT. Model dilatih dengan BATCH\_SIZE bernilai 64, yang artinya sebanyak 64 sampel data secara serentak diproses pada jaringan saraf didalam model *training*, kemudian adalah EPOCHS yang bernilai 1000, yang artinya setiap parameter pada jaringan saraf didalam model LSTM diperbarui sebanyak 1000 kali.



### C. Hasil Prediksi

Tabel 1. Hasil prediksi harga saham MSFT secara harian

Tabel 2. Hasil prediksi harga saham BBKA secara harian

TANGGAL DATA SAHAM	PREDIKSI PADA HARI KE-X	TANGGAL YANG PREDIKSIKA	HARGA CLOSING SEBELUM-NYA	PREDIKSI HARGA CLOSING	HARGA CLOSING REAL	PREDIKSI PERGERAKAN KAN	PERGERAKAN REAL	NILAI AKURASI PREDIKSI
14 Juli 2022	1	15 Juli 2022	Rp7.025,00	Rp7.023,00	Rp7.000,00	▼0,03%	▼0,36%	99,7%
15 Juli 2022	1	18 Juli 2022	Rp7.000,00	Rp7.093,00	Rp7.150,00	▲1,31%	▲2,10%	99,2%
18 Juli 2022	1	19 Juli 2022	Rp7.150,00	Rp7.045,00	Rp7.175,00	▼1,49%	▲0,35%	98,2%
19 Juli 2022	1	20 Juli 2022	Rp7.175,00	Rp7.067,77	Rp7.400,00	▼1,52%	▲3,04%	95,5%
20 Juli 2022	1	21 Juli 2022	Rp7.400,00	Rp7.301,04	Rp7.400,00	▼1,36%	►0,00%	98,7%
21 Juli 2022	1	22 Juli 2022	Rp7.400,00	Rp7.339,76	Rp7.325,00	▼0,82%	▼1,02%	99,8%
22 Juli 2022	1	25 Juli 2022	Rp7.325,00	Rp7.286,27	Rp7.300,00	▼0,53%	▼0,34%	99,8%
							Rata-Rata	98,7%

Tabel 3. Hasil Prediksi Harga Saham BBCA  
Dalam Rentang Waktu Berbeda

TANGGAL DATA SAHAM TERAKHIR	PREDIKSI PADA HARI KE-X	TANGGAL YANG DIPREDIKSIKAN	HARGA CLOSING SEBELUM-NYA	PREDIKSI HARGA CLOSING	HARGA CLOSING REAL	PREDIKSI PERGERAKAN	PERGERAKAN REAL	NILAI AKURASI PREDIKSI
11 Juli 2022	1	12 Juli 2022	Rp7.125	Rp7.206	Rp7.175	▲1,12%	▲0,70%	99,6%
11 Juli 2022	7	20 Juli 2022	Rp7.125	Rp7.149	Rp7.400	▲0,34%	▲3,72%	96,6%
12 Juli 2022	10	26 Juli 2022	Rp7.175	Rp7.135	Rp7.300	▼0,56%	▲1,71%	97,7%
12 Juli 2022	15	02 Agustus 2022	Rp7.175	Rp7.104	Rp7.600	▼1,00%	▲5,59%	93,5%
12 Juli 2022	20	09 Agustus 2022	Rp7.175	Rp6.996	Rp7.900	▼2,56%	▲9,18%	88,6%
Rata-Rata								95,2%

Tabel 4. Hasil Prediksi Harga Saham MSFT  
Dalam Rentang Waktu Berbeda

TANGGAL DATA SAHAM TERAKHIR	PREDIKSI PADA HARI KE-X	TANGGAL YANG DIPREDIKSIKAN	HARGA CLOSING SEBELUM-NYA	PREDIKSI HARGA CLOSING	HARGA CLOSING REAL	PREDIKSI PERGERAKAN	PERGERAKAN REAL	NILAI AKURASI PREDIKSI
13 Juli 2022	1	14 Juli 2022	\$252,72	\$254,11	\$254,08	▲0,55%	▲0,54%	100,0%
13 Juli 2022	7	22 Juli 2022	\$252,72	\$262,24	\$262,27	▲3,63%	▲3,64%	100,0%
13 Juli 2022	10	27 Juli 2022	\$252,72	\$253,71	\$268,74	▲0,39%	▲5,95%	94,4%
13 Juli 2022	15	03 Agustus 2022	\$252,72	\$265,19	\$282,74	▲4,70%	▲10,62%	93,8%
13 Juli 2022	20	10 Agustus 2022	\$252,72	\$232,88	\$289,16	▼8,52%	▲12,60%	80,5%
Rata-Rata								93,7%

Dapat dilihat bahwa nilai akurasi dari masing-masing prediksi memiliki trend penurunan seiring bertambahnya hari yang diprediksi, peneliti berpendapat bahwa kemungkinan terbesar mengapa nilai akurasi mengalami penurunan karena jarak data dengan target prediksi yang terlalu jauh, mengingat bahwa harga saham sangat mudah berubah setiap harinya dikarenakan berbagai faktor, mulai dari faktor politik, sosial, ekonomi, dan faktor-faktor internal pada perusahaan. Setelah mendapatkan seluruh akurasi hasil prediksi tersebut, kemudian dilakukan perhitungan rata-rata keseluruhan yang didapatkan dengan perhitungan sebagai berikut :

$$R_t = \frac{\bar{x}}{n} \times 100\% \quad (8)$$

$$R_t = \frac{30.2}{31} \times 100\%$$

$$R_t = 0.974 \times 100\%$$

$$R_t = 97,4\%$$

Jadi, rata-rata keseluruhan akurasi prediksi adalah 97,4%

## PENUTUP

### A. Kesimpulan

Kesimpulan dari hasil penelitian dan pengembangan rancang bangun aplikasi prediksi harga saham menggunakan *deep learning* dengan algoritma *long short-term memory* yaitu :

1. Cara membangun suatu aplikasi prediksi harga saham menggunakan *deep learning* dengan algoritma *long short-term memory* adalah sebagai berikut :

- a. Mengumpulkan data perdagangan saham menggunakan API Yahoo Finance.
- b. Data yang telah diambil atau dikumpulkan kemudian diskalakan menjadi skala 0-1 dan dibuat dua *dataset*, yaitu *dataset* latih dan *dataset* test.
- c. Membuat model LSTM.
- d. Melatih model dengan kedua *dataset*.
- e. Proses prediksi dilakukan dengan mengambil model hasil latihan kemudian menggunakan beberapa data perdagangan terakhir seperti 80 hari terakhir untuk menentukan harga saham kedepannya.
- f. Hasil prediksi kemudian dikirim ke *client-side* Anvil Works agar pengguna dapat melihat hasil prediksi dengan mudah.

### B. Saran

Saran yang dapat disampaikan untuk aplikasi prediksi harga saham menggunakan *deep learning* dengan algoritma *long short-term memory* untuk kedepannya adalah :

- a. Diperlukannya training dengan hardware spesifikasi tinggi, hardware yang memiliki tingkat komputasi cepat untuk menyingkat waktu latihan model prediksi.
- b. Dengan lebih cepatnya waktu latihan model prediksi, maka dapat dilakukan lebih banyak latihan model prediksi untuk berbagai saham lain yang akan memudahkan para pengguna mengetahui hasil prediksi dari lebih banyak saham.
- c. Dengan lebih cepatnya waktu latihan model prediksi, maka dapat dilakukan lebih banyak percobaan untuk meningkatkan akurasi dari prediksi yaitu dengan meningkatkan jumlah epochs dan jumlah *batch\_size*.
- d. Dengan menggunakan hardware yang memiliki tingkat komputasi lebih cepat maka dapat dilakukan lebih banyak tuning dengan cara merubah beberapa parameter latihan dan prediksi untuk melihat apakah hasil prediksi lebih akurat.

## DAFTAR PUSTAKA

- [1] Aprian, B. A. (2020). *PREDIKSI PENDAPATAN CARGO SERVICE CENTER TANGERANG CITY MENGGUNAKAN ARSITEKTUR LONG SHORT TERM MEMORY*. Universitas Muhammadiyah Malang.  
<https://eprints.umm.ac.id/62460/>
- [2] Bursa Efek Indonesia (BEI). (2010). *Definisi Saham*.
- [3] Dolphin, R. (2020). *Long Short-Term Memory*.  
<https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9>
- [4] IBM. (2020). *Deep Learning*.  
<https://www.ibm.com/cloud/learn/deep-learning#:~:text=Deep learning attempts to mimic,make predictions with incredible accuracy>.