

# PREDIKSI HARGA FOREIGN EXCHANGE MATA UANG EUR/USD DAN GBP/USD MENGGUNAKAN LONG SHORT-TERM MEMORY

Angel Joanna Wijaya<sup>1</sup>, Windra Swastika<sup>2</sup>, Oesman Hendra Kelana<sup>3</sup>

Program Studi Teknik Informatika, Universitas Ma Chung

Email : [311710002@student.machung.ac.id](mailto:311710002@student.machung.ac.id) , [windra.swastika@machung.ac.id](mailto:windra.swastika@machung.ac.id) ,  
[oesman.hendra@machung.ac.id](mailto:oesman.hendra@machung.ac.id)

## Abstrak

Foreign exchange (Forex) adalah perdagangan pasangan mata uang dari harga mata uang suatu negara terhadap mata uang negara lainnya. Pada penelitian ini menggunakan metode Long Short-Term Memory (LSTM) untuk memprediksi harga close mata uang EUR/USD (Euro terhadap Dolar Amerika) dan GBP/USD (Pound Sterling terhadap Dolar Amerika) pada candle D1 (1 hari) dengan input harga open dan close. Hasil yang diperoleh model EUR/USD dengan 1 input mendapatkan nilai Mean Squared Error (MSE) terendah yaitu 0,0535 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Nadam. Pada model 3 input mendapatkan nilai MSE terendah yaitu 0,0529 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Nadam. Sedangkan, model 5 input mendapatkan nilai MSE terendah yaitu 0,0469 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Nadam. Pada mata uang GBP/USD, model dengan 1 input mendapatkan nilai MSE terendah yaitu 0,0543 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Nadam. Pada model 3 input mendapatkan nilai MSE terendah yaitu 0,0520 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Adam. Sedangkan, model 5 input mendapatkan nilai MSE terendah yaitu 0,0631 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Adam. Model dengan MSE terbaik yang digunakan pada website yang dibuat. Rekomendasi yang diberikan adalah pada mata uang EUR/USD menggunakan model 5 input dan mata uang GBP/USD menggunakan model 3 input. Hasil loss MSE, akurasi MSE, dan jumlah profit yang didapatkan semuanya adalah yang terbaik jika dibandingkan dengan model dengan jumlah input lainnya.

**Kata kunci:** LSTM, RNN, Forex

## Abstract

Foreign exchange (Forex) is trading of currency pairs from price of one country's currency against another country's currency. In this study, Long Short-Term Memory (LSTM) method is used to predict close price of EUR/USD (Euro against US Dollar) and GBP/USD (Pound Sterling against US Dollar) on candle D1 (1 day) with the input open price and close price. The results obtained by the EUR/USD model with 1-input get the lowest Mean Squared Error (MSE) value of 0.0535 with 10-node 1-layered LSTM model and using the Nadam optimizer. In the 3-input model, the lowest MSE value is 0.0529 with 10-node 1-layered LSTM model and uses the Nadam optimizer. Meanwhile, 5-input model gets the lowest MSE value, which is 0.0469 with a 10-node 1-layered LSTM model and uses the Nadam optimizer. In GBP/USD currency, model with 1-input gets the lowest MSE value of 0.0543 with 10- node 1 layer LSTM model and uses the Nadam optimizer. In 3 input model, the lowest MSE value is 0.0520 with 10-node 1-layered LSTM model and uses the Adam optimizer. Meanwhile, 5-input model gets the lowest MSE value, which is 0.0631 with 10-node 1-layered LSTM

model and uses the Adam optimizer. Model with the best MSE used on the website created. The recommendation given is that the EUR/USD currency uses a 5-input model and the GBP/USD currency uses a 3-input model. The results of MSE loss, MSE accuracy, and the amount of profit obtained are all the best when compared to models with other input.

**Keywords:** LSTM, RNN, Forex

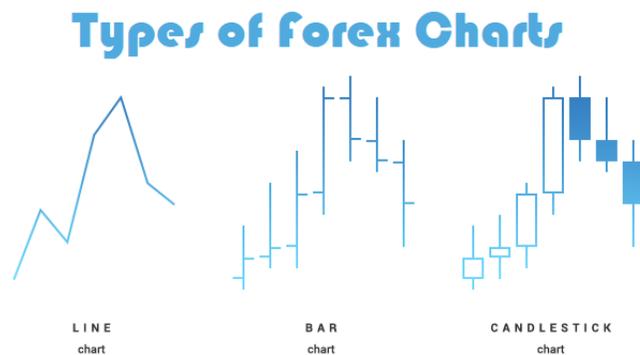
## Pendahuluan

Investasi adalah penempatan modal pada suatu aset/perusahaan dengan tujuan mendapatkan keuntungan dalam waktu jangka panjang. Salah satu bidang investasi adalah Foreign Exchange atau yang biasa disebut Forex. Forex merupakan perdagangan pasangan mata uang dari harga mata uang suatu negara terhadap mata uang negara lainnya. Penentu naik turunnya harga tersebut adalah jumlah penawaran dan permintaan terhadap ekspor-impor yang akan mempengaruhi nilai tukar suatu mata uang dalam perdagangan internasional (William, Hidayatno, dan Zahra, 2014). Permasalahannya di Indonesia adalah investasi masih sering tidak dianggap penting oleh masyarakat Indonesia. Investasi memang sangat penting, namun tidak sedikit yang kehilangan uang/mengalami kerugian besar saat berinvestasi, pada bidang Forex khususnya. Investasi di bidang Forex memang memiliki potensi keuntungan yang tinggi, namun risikonya juga tinggi. Metode yang akan digunakan pada penelitian ini adalah salah satu varian algoritma Recurrent Neural Network (RNN) yaitu Long Short-Term Memory (LSTM). Algoritma RNN merupakan algoritma yang cocok untuk diimplementasikan pada data yang bersifat sequence atau berurutan seperti time series. LSTM (Long Short-Term Memory) merupakan jenis modul pemrosesan lain untuk RNN yang diciptakan oleh Hochreiter & Schmidhuber (1997). LSTM network terdiri dari beberapa LSTM cell yang mempunyai sistem pemrosesan yang berbeda dengan modul RNN biasa. Penelitian terdahulu yaitu oleh Dautel, Hardle, Lessmann, dan Seow (2020) melakukan penelitian dengan membandingkan model FNN (Feedforward Neural Network), SRNN (Simple Recurrent Neural Network), LSTM (Long Short-Term Memory), dan GRU (Gated Recurrent Unit). Percobaan ini dilakukan pada 4 mata uang yaitu EUR/USD, GBP/USD, USD/JPY, dan USD/CHY. Hasil yang didapatkan adalah LSTM dan GRU mendapatkan hasil yang lebih baik

dibandingkan FNN dan SRNN. Pada tahun 2021, penelitian oleh Yildirim, Toroslu dan Fiore mengusulkan model *hybrid* yang terdiri dari model LSTM makroekonomi dan LSTM teknikal. Penelitian ini memperkirakan arah dari pasangan mata uang EUR/USD dengan *output* klasifikasi dari 3 kelas yaitu *no action*, *decrease*, dan *increase*. Evaluasi model menggunakan *profit accuracy*, hasil yang didapatkan adalah rata – rata 73,61%. Pada penelitian ini, data yang digunakan adalah data harga *open* (pembukaan), *high* (tertinggi), *low* (terendah), dan *close* (penutupan) pada *candle* D1 (1 hari) pada pasangan mata uang EUR/USD (Euro terhadap Dolar Amerika) dan GBP/USD (Pound Sterling terhadap Dolar Amerika). Prediksi harga *close* pada hari ini dengan mencoba berbagai *input* yaitu 1 *input*, 3 *input*, dan 5 *input*. Alasan pemilihan mata uang EUR/USD dan GBP/USD adalah keduanya termasuk dalam pasangan mata uang *major pairs*. EUR/USD memiliki tingkat *volatility* yang cenderung rendah sehingga *liquidity*-nya tinggi dan pergerakannya stabil. Sedangkan GBP/USD memiliki tingkat *volatility* yang cenderung tinggi sehingga *liquidity*-nya rendah dan pergerakannya kurang stabil. Peneliti akan menguji metode LSTM pada 2 mata uang *major pairs* yang memiliki tingkat *volatility* rendah dan tinggi.

### Tinjauan Pustaka

Foreign Exchange atau yang biasa disebut Forex adalah perdagangan pasangan mata uang dari harga mata uang suatu negara terhadap mata uang negara lainnya (pasangan mata uang atau pair) yang melibatkan pasar – pasar uang utama di dunia selama 24 jam secara berkesinambungan (Sihombing, 2013). Terdapat 3 macam pair yaitu pasangan mata uang mayor (major currency pairs), pasangan mata uang minor (minor currency pairs) atau yang sering disebut juga cross currency pairs, dan pasangan mata uang eksotik (exotic currency pairs). Pada penelitian ini menggunakan pasangan mata uang mayor yaitu EUR/USD (Euro terhadap US Dollar) dan GBP/USD (Pound Sterling terhadap US Dollar). Terdapat 3 *charts* yang sering digunakan pada Forex yaitu *line chart*, *bar chart*, dan *candlestick chart* (gambar 1). *Line chart* adalah *chart* yang paling mudah untuk dibaca saat melihat pergerakan harga. Garis harga pada *line chart* adalah garis dari antara 2 harga penutupan (*close*). Saat garis tersebut semua dihubungkan maka dapat dilihat bagaimana perubahan harga selama jangka waktu tertentu. *Bar chart* lebih rumit dibandingkan *line chart*. Pada *bar chart*, dapat dilihat harga pembukaan (*open*), harga tertinggi (*high*), harga terendah (*low*), harga penutupan (*close*). *Candlestick chart* adalah *chart* yang mirip dengan *bar chart*, hanya berbeda bentuk saja. Pada *candlestick chart* dapat dilihat harga pembukaan (*open*), harga tertinggi (*high*), harga terendah (*low*), harga penutupan (*close*). *Candlestick chart* adalah *chart* yang paling sering digunakan trader.



Gambar 1. Forex Charts

(sumber: <https://www.forexbrokerslab.com/forex-trading-chart/>)

Gambar 2 merupakan anatomi *candle* pada *candlestick chart*. *Bullish candle* adalah *candle* yang menunjukkan kondisi *uptrend*, di mana harga *close* lebih tinggi dari harga *open*. Sedangkan *bearish candle* adalah *candle* yang menunjukkan kondisi *downtrend*, di mana harga *close* lebih rendah dari harga *open*. Pada dasarnya *bullish candle* berwarna putih namun ada juga yang berwarna hijau seperti pada gambar. Sedangkan *bearish candle* berwarna hitam dan bisa juga berwarna merah. Pada gambar 2 juga ditunjukkan OHLC pada setiap *candle*. OHLC adalah singkatan dari *open*, *high*, *low*, *close*. Pada setiap *candle* mewakili harga dari jangka waktu (*time frame*) tertentu. Harga *open* muncul saat pertama kali *candle* tersebut dibuat, sedangkan harga *close* muncul saat terakhir *candle* akan berpindah ke *candle* yang baru. *High* dan *low* muncul saat harga tidak ada yang lebih tinggi dan lebih rendah pada *candle* tersebut.



Gambar 2 Candle Anatomy

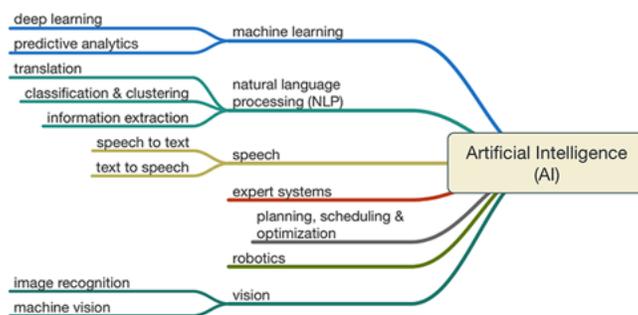
(sumber: <https://erronjohn.blogspot.com/2020/10/apa-itu-ohlc.html>)

*Time frame* adalah periode atau kurun waktu pergerakan yang ditampilkan oleh *chart*. *Time frame* dalam MetaTrader (aplikasi untuk *trading*) umumnya ditampilkan dalam kurun waktu 1 menit (M1), 5 menit (M5), 15 menit (M15), 30 Menit (M30), 1 Jam (H1), 4 Jam (H4), 1 Hari (D1), Mingguan (W1), dan Bulanan (MN). Pada *candle* dengan *time frame* M1, maka harga pada setiap 1 *candle* merupakan pergerakan harga dari rentang waktu 1 menit, pada *time frame* M5 harga pada setiap 1 *candle* merupakan pergerakan harga dari rentang waktu 5 menit, dan begitu seterusnya. *Time frame* ini disediakan untuk *trader* agar bisa melihat pergerakan harga dari beberapa rentang waktu sehingga bisa melihat pergerakan sesungguhnya dan meningkatkan kejelian *trader*. Pada penelitian ini, akan menggunakan *time frame* D1 yaitu *candle* 1 hari.

Trading Forex pada prinsipnya adalah pembelian simultan dari sebuah mata uang dan penjualan mata uang lainnya. Pasangan mata uang tersebut kemudian diperdagangkan secara bersamaan. Contohnya, harga EUR/USD = 1.2340, maka mata uang yang berada di sisi kiri (Euro) adalah mata uang dasar, sedangkan mata uang di sisi kanan disebut mata uang kutipan (US Dollar). Harga EUR/USD = 1.2340 menunjukkan bahwa 1 unit mata uang dasar (Euro) adalah senilai 1.23400 mata uang kutipan (USD). Sehingga artinya 1 Euro adalah senilai 1.23400 US Dollar.

Dalam trading Forex, terdapat 2 opsi yaitu *buy* (beli) atau *sell* (jual). *Trader* tidak perlu membeli dahulu baru bisa menjual, bahkan *trader* bisa membuka transaksi jual dan beli secara bersamaan. Selain *buy* dan *sell* yang melakukan transaksi dengan harga saat itu juga, *trader* juga bisa membuat *pending order* yang dinamakan *buy / sell stop* dan *buy / sell limit*. *Buy stop* diterapkan jika *trader* ingin membuka posisi *buy* di atas harga pasar saat ini. Sedangkan *sell stop* diterapkan jika *trader* ingin membuka posisi *sell* di bawah harga pasar sekarang. *Buy stop* dan *sell stop* tidak akan aktif jika harga belum sampai ke level yang telah *trader* tentukan. Lalu, *buy limit* diterapkan jika *trader* ingin membuka posisi *buy* di bawah harga pasar saat ini dan *sell limit* diterapkan jika *trader* ingin membuka posisi *sell* di atas harga pasar saat ini. Setelah *trader* membuka posisi, *trader* bisa juga langsung mengatur *take profit* (TP). Saat harga menyentuh TP tersebut, maka *trader* mendapatkan keuntungan. Selain TP, *trader* juga bisa mengatur *stop loss* (SL). SL digunakan untuk membatasi kerugian yang muncul akibat harga yang bergerak tidak sesuai dengan ekspektasi *trader*. Saat pergerakan harga menyentuh nilai ini, maka secara otomatis akan menutup *order* atau posisi tersebut.

*Artificial Intelligence* (AI) atau Kecerdasan Buatan adalah salah satu bagian dari ilmu komputer yang mempelajari tentang bagaimana membuat mesin atau komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia bahkan lebih baik daripada yang dilakukan manusia (Dahria, 2008). AI merupakan teknologi yang memerlukan data untuk dijadikan pengetahuan/pembelajaran, sama seperti manusia. AI membutuhkan pengalaman dan pembelajaran dari data agar kecerdasannya bisa bertambah baik. Poin penting dalam proses AI adalah *learning*, *reasoning* dan *self-correction*. Proses belajar AI-pun tidak selalu disuruh oleh manusia, AI bisa belajar dengan sendirinya berdasarkan pengalamannya saat digunakan oleh manusia. Contoh aplikasinya adalah *Self Driving Cars* dan *Sign and Voice Recognizer* (SVR). Pada gambar 3, cabang ilmu *Artificial Intelligence* yaitu *Machine Learning*, *Natural Language Processing* (NLP), *Speech*, *Expert Systems*, *Planning*, *Scheduling*, dan *Optimization*, *Robotics*, dan *Vision*. Pada penelitian ini menggunakan *Deep Learning* yang merupakan cabang dari *Machine Learning*.



Gambar 3. Cabang Artificial Intelligence

(sumber:

<https://medium.com/@shivakumar.goniwada/microservices-architecture-in-artificial-intelligence-60bac5b4485d>)

Menurut Dahria (2008), terdapat perbedaan dari kecerdasan buatan dan pemrograman konvensional, yaitu seperti pada tabel 1.

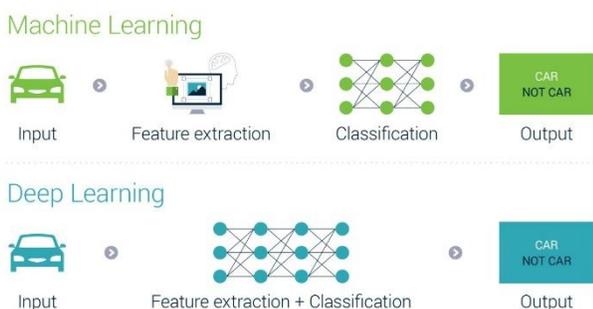
Tabel 1. Kecerdasan Buatan VS Pemrograman Konvensional

Dimensi	Kecerdasan Buatan	Pemrograman Konvensional
Pemrosesan	Pengetahuan diperoleh dari mekanisme pemrosesan (inferensi)	Digabung dalam satu program sekuensial
Eksekusi	Eksekusi dilakukan secara heuristik dan logis	Secara algoritma ( <i>step-by-step</i> )
Sifat Input	Bisa tidak lengkap	Harus lengkap
Manipulasi	Efektif pada basis pengetahuan yang besar	Efektif pada <i>database</i> yang besar
Keterangan (penjelasan)	Disediakan	Biasanya tidak disediakan
Fokus	Pengetahuan	Data dan informasi
Struktur	Kontrol dipisahkan dari pengetahuan	Kontrol terintegrasi dengan informasi (data)
Sifat Output	Kualitatif	Kuantitatif
Perubahan	Perubahan pada kaidah dapat dilakukan dengan kaidah yang sedikit	Pada program merepotkan
Kemampuan menalar	Ya	Tidak

“*Machine Learning* (ML) is a subset of *Artificial Intelligence* (AI) technique which use statistical methods to enable machines to improve with experience”, Garg, 2019. *Machine Learning* memberikan kemampuan kepada sistem komputer untuk belajar secara mandiri menggunakan data yang disediakan dan membuat prediksi yang akurat. Data adalah komponen yang penting dalam *Machine Learning*. Data akan diproses oleh algoritma dan metode yang berbeda pada *Machine Learning* melalui berbagai percobaan untuk memperoleh hasil yang optimal. Pada *Machine Learning* data

akan dibagi menjadi 2 yaitu data *training* dan data *testing*. Data *training* digunakan untuk melakukan pelatihan data pada algoritma dalam mencari model yang cocok, sedangkan data *testing* digunakan untuk melihat performa dari model yang sudah dilatih sebelumnya dan menemukan data baru yang belum pernah ditemukan sebelumnya.

*Deep Learning* merupakan salah satu cabang bidang *Machine Learning* yang mengadopsi cara kerja otak manusia agar mesin atau komputer bisa belajar secara mandiri. *Deep Learning* termasuk cabang *Machine Learning* jenis *Supervised Learning*. *Deep learning* dirancang untuk menganalisis data dengan mengikuti atau menduplikasi cara kerja otak manusia dalam mengambil keputusan (Peryanto et al., 2019).



**Gambar 4 Machine Learning VS Deep Learning**

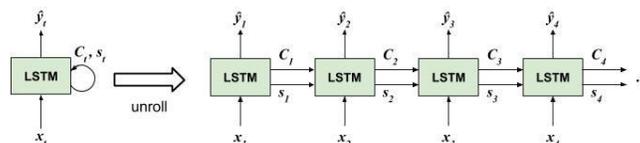
(sumber: <https://www.inteliment.com/blog/our-thinking/lets-understand-the-difference-between-machine-learning-vs-deep-learning/>)

Pada gambar 4 adalah struktur pemodelan jaringan pada *Machine Learning* dan *Deep Learning*. Terdapat perbedaan pada saat tahap *feature extraction*. Pada *Machine Learning* *feature extraction* dilakukan dengan bantuan manusia, namun pada *Deep Learning* *feature extraction* dilakukan oleh komputer secara mandiri. *Feature extraction* adalah proses reduksi dimensi di mana kumpulan data mentah awal direduksi menjadi grup yang lebih mudah dikelola untuk diproses. *Feature extraction* adalah nama untuk metode yang memilih dan/atau menggabungkan variabel ke dalam fitur, secara efektif mengurangi jumlah data yang harus diproses, sambil tetap mendeskripsikan kumpulan data asli secara akurat dan lengkap. Proses ini berguna saat mengurangi jumlah *resource* yang diperlukan untuk pemrosesan tanpa kehilangan informasi yang penting dan relevan. *Feature extraction* juga dapat mengurangi jumlah data yang berlebihan untuk analisis tertentu. Selain itu, pengurangan data dalam membuat kombinasi variabel (fitur) dapat mempercepat proses pembelajaran dan langkah-langkah generalisasi dalam proses pembelajaran mesin.

*Neural Network* atau Jaringan Syaraf Tiruan (JST) terinspirasi dari cara kerja neuron pada otak manusia. Menurut Widiputra (2016), *Neural Network* merupakan sebuah teknik atau pendekatan pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam memproses informasi.

*Recurrent Neural Network* (RNN) adalah jenis *Neural Network*

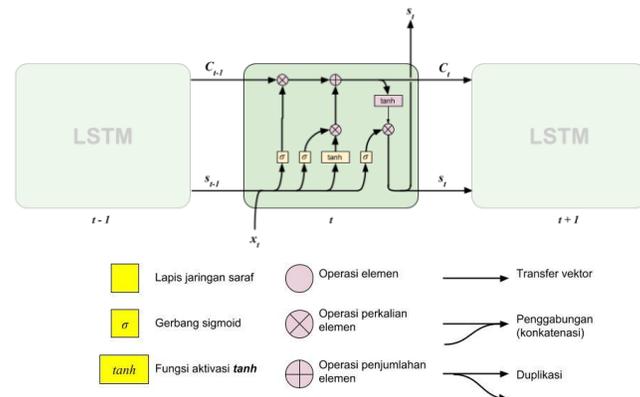
yang pemrosesannya berulang kali dipanggil untuk memproses *input* data sekuensial. RNN melakukan pengulangan dengan memanfaatkan data masa lalu. Perbedaan RNN dibandingkan dengan NN biasa adalah pada RNN *output* dari *node* sebelumnya dipakai sebagai *input* bersamaan dengan *input* pada *node* tersebut.



**Gambar 5. LSTM Network**

(sumber: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/>)

LSTM (*Long Short-Term Memory*) merupakan jenis modul pemrosesan lain untuk RNN yang diciptakan oleh Hochreiter & Schmidhuber (1997). LSTM network terdiri dari beberapa LSTM *cell* yang mempunyai sistem pemrosesan yang berbeda dengan modul RNN biasa. Setiap 1 unit LSTM disebut sebuah LSTM *cell*, sedangkan dalam LSTM network terdiri dari banyak LSTM *cells*. Gambar 5 adalah contoh dari LSTM network yang terdiri dari banyak LSTM *cells*.

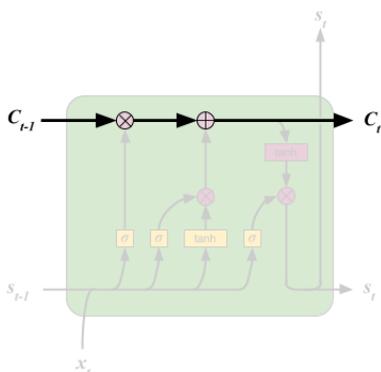


**Gambar 6. LSTM Cells**

(sumber: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/>)

Pada gambar 6, merupakan arsitektur LSTM *cells*. Setiap garis membawa seluruh vektor, dari satu simpul (*node*) ke simpul yang lain. Lingkaran merah muda mewakili operasi elemen, seperti penambahan atau perkalian elemen vektor, sedangkan kotak kuning adalah *layer neural network* (mengandung parameter dan bias) yang bisa belajar. Dua garis yang tergabung menjadi satu artinya penggabungan dua matriks/vektor, sementara garis berpisah menandakan duplikasi dan salinannya pergi ke simpul yang berbeda.

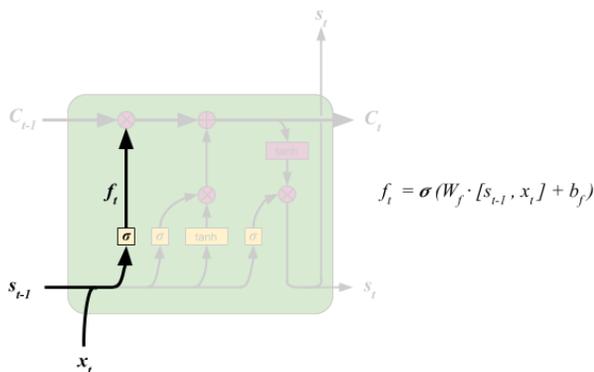
Arsitektur LSTM *cells* dari beberapa lajur yaitu lajur *memory*, lajur *forget*, lajur *new knowledge creation*, dan lajur *output*.



**Gambar 7. Lajur Memory**

(sumber: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/>)

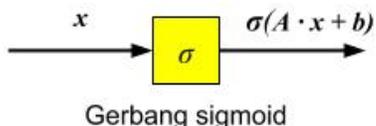
Pada gambar 7, konteks  $C_t$  disebut juga *cell state* atau *memory cell*. Dengan adanya lajur *memory*, suatu nilai di konteks yang lama akan dengan mudah diteruskan ke konteks yang baru dengan sedikit modifikasi (jika diperlukan). Konteks adalah sebuah vektor, yang jumlah elemennya ditentukan sendiri.



**Gambar 8. Lajur Forget**

(sumber: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/>)

Pada gambar 8,  $f_t$  merupakan gerbang *sigmoid* yang disebut gerbang lupa atau *forget gate*. Gerbang ini membaca nilai  $s_{t-1}$  dan  $x_t$ , dan menghasilkan angka antara 0 dan 1 untuk setiap elemen dalam  $C_{t-1}$ . Nilai 1 artinya nilai disimpan, sedangkan 0 artinya nilai dibuang atau diabaikan.

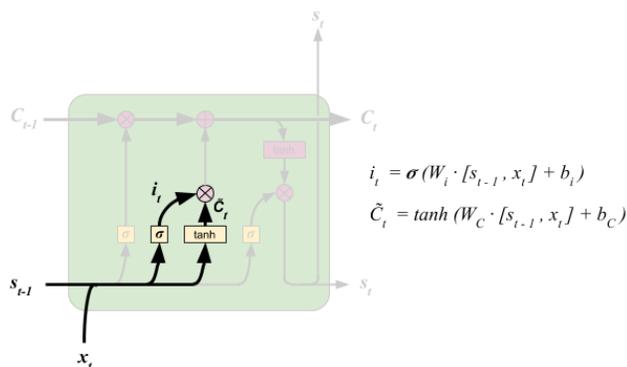


**Gambar 9. Gerbang Sigmoid**

(sumber: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/>)

Pada gambar 9, untuk suatu *input*  $x$ , *output* dari gerbang *sigmoid* adalah  $\sigma(A \cdot x + b)$ , di mana  $A$  adalah parameter,  $x$

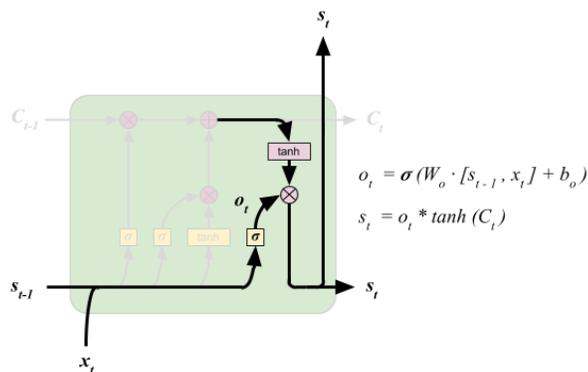
adalah *input*, dan  $b$  adalah bias, keduanya dipelajari dalam proses latihan, dan  $\sigma$  adalah fungsi *sigmoid*. Keluaran gerbang *sigmoid* adalah angka antara 0 dan 1.



**Gambar 10. Lajur New Knowledge Creation**

(sumber: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/>)

Pada lajur *new knowledge creation* (gambar 10), bagian ini memutuskan informasi baru apa yang akan digunakan pada  $C_t$ . Proses ini memiliki dua bagian. Pertama, gerbang *sigmoid* yang disebut gerbang masukan atau *input gate* ( $i_t$ ) memutuskan nilai mana yang akan diperbaharui. Lalu sebuah layer *tanh* menghasilkan kandidat vektor konteks baru,  $\hat{C}_t$ . Lalu digabungkan keduanya untuk membuat pembaharuan ke konteks selanjutnya.



**Gambar 11. Lajur Output**

(sumber: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/>)

Pada gambar 11,  $O_t$  merupakan gerbang *sigmoid* yang dinamakan gerbang *output* atau *output gate*. *Output gate* digunakan untuk memutuskan bagian – bagian apa dari konteks yang akan dihasilkan. Saat konteks melalui *tanh*, hasil yang didapatkan nilainya antara  $-1$  dan  $1$ , setelah itu dikalikan dengan *output gate*.

Python merupakan bahasa pemrograman tingkat tinggi berorientasi objek yang diciptakan oleh Guido Van Rossum pada tahun 1990 di Stichting Mathematisch Centrum (CWI), Amsterdam. Pada tahun 1995 Python dikembangkan lagi agar lebih kompatibel. Selanjutnya pada awal tahun 2000, terdapat

pembaharuan versi Python hingga mencapai versi 3 sampai saat ini. Bahasa Python dibuat untuk menggantikan bahasa pemrograman ABC. Python telah menjadi bahasa pemrograman yang populer dan banyak digunakan oleh *Data Analysts*, *Data Scientists*, dan para *Software Engineers* untuk pengolahan data pada *Machine Learning*. Salah satu kelebihan Python yaitu bahasa ini dilengkapi banyak fitur yaitu *library* yang dapat digunakan untuk komputasi *Machine Learning* maupun *Deep Learning*. Banyak orang tertarik untuk menggunakan Python karena dianggap mudah untuk dipelajari, sekalipun untuk pemula. Kode-kode yang ada di dalamnya mudah dibaca dan dapat menjalankan banyak fungsi kompleks dengan mudah karena banyaknya *library* yang disediakan. Pengembangan program pada Python-pun dapat dilakukan dengan cepat dan menggunakan lebih sedikit kode. Bahkan Python mampu menjadikan program dengan skala sangat rumit menjadi mudah dengan bantuan *library* tersebut. Python sendiri mendukung *multi-platform* dan *multi system* serta memiliki sistem pengelolaan memori otomatis seperti Java.

NumPy yang merupakan kependekan dari Numerical Python adalah salah satu *library* teratas yang dilengkapi dengan sumber daya yang berguna untuk membantu para *Data Scientist* mengubah Python menjadi alat analisis dan pemodelan ilmiah yang kuat (Purwanto, 2018). Menurut Rohman (2019), NumPy adalah *library* Python yang berfokus pada *scientific computing*. NumPy memiliki kemampuan untuk membentuk objek *N-dimensional array* yang mirip dengan *list* pada Python. Keunggulan NumPy *array* dibandingkan dengan *list* pada Python adalah penggunaan *memory* yang lebih kecil serta *runtime* yang lebih cepat. NumPy juga memudahkan melakukan komputasi pada aljabar linear, terutama operasi pada vektor (*1-dimension array*) dan matriks (*2-dimension array*)

Pandas (*Python for Data Analysis*) adalah sebuah *library* berlisensi BSD dan *open source* yang menyediakan struktur data dan analisis data yang mudah digunakan dan berkinerja tinggi untuk bahasa pemrograman Python (Mutmainnah, 2019). Menurut Rohman (2019), Pandas adalah *library* Python yang fokus untuk proses menganalisis data seperti manipulasi data, persiapan data, dan pembersihan data. Pandas biasanya digunakan untuk membaca dan membersihkan data mentah ke dalam sebuah format khusus sebelum siap untuk dianalisis. Pandas bisa digunakan untuk menyelaraskan data untuk perbandingan dan penggabungan set data, penanganan data yang hilang, dan lain – lain. Awal mula dibuat *library* Pandas adalah untuk menangani data finansial.

TensorFlow adalah platform *open source end-to-end* yang dirancang untuk komputasi numerik khususnya membangun model *Machine Learning* seperti *neural network*. TensorFlow dikembangkan oleh tim Google Brain untuk penggunaan internal Google dan dirilis di bawah lisensi *open source Apache 2.0* pada 9 November 2015. Komponen inti TensorFlow adalah *tensor* dan grafik komputasi yang melintasi *node* hingga *edge* (Gifari, 2020). *Tensor* adalah objek matematika yang digunakan untuk mendeskripsikan properti fisik seperti skalar dan vektor. Pada dasarnya, *tensor* hanyalah sebuah generalisasi

dari skalar dan vektor; skalar adalah *tensor* peringkat nol, sedangkan vektor adalah *tensor* peringkat pertama. Sedangkan grafik komputasi adalah entitas yang didefinisikan sebagai grafik berarah di mana setiap *node* sesuai dengan operasi matematika tertentu. Pada dasarnya, grafik komputasi adalah alat untuk mengekspresikan dan mengevaluasi beberapa ekspresi matematis pada setiap *node*.

Keras adalah *library open source* yang ditulis dalam bahasa Python yang dirancang untuk menyederhanakan model *Deep Learning*. Tujuan utama dibuatnya Keras ini adalah untuk membuat prototipe dari *Neural Network* secara cepat. Keras dapat dijalankan pada *framework* kecerdasan buatan seperti TensorFlow, Microsoft Cognitive Toolkit, dan Theano (Fachrizal, 2019). *Library* ini dikembangkan oleh François Chollet sebagai bagian dari *research project* dan dirilis sebagai proyek *open source* pada Maret 2015. Keras dengan cepat mendapatkan popularitas karena keramahan atau kemudahan penggunaan, fleksibilitas, dan desainnya yang indah. Sebagian besar fungsi Keras seperti *losses*, *regularizers*, *constraints*, *initializers*, *metrics*, *activation functions*, *layers*, dan bahkan model lengkap dapat disesuaikan dengan cara yang sama. Sering kali, hanya perlu menulis *simple function*, dengan *input* dan *output* yang sesuai (Aurélien Géron, 2019).

Keras menyediakan banyak fungsi *built-in* terkait *Neural Network* untuk membuat model Keras dan lapisan Keras. Beberapa fungsi tersebut adalah sebagai berikut:

1. *Activation function*: fungsi aktivasi adalah konsep penting dalam *Neural Network* dan modul aktivasi ini menyediakan banyak fungsi aktivasi seperti Softmax dan ReLU
2. *Loss function*: menyediakan fungsi-fungsi *loss* seperti *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), Poisson, dan lain-lain
3. *Optimize*: menyediakan fungsi-fungsi optimizer seperti Adam, SGD, dan lainnya
4. *Regularizers*: menyediakan fungsi-fungsi Regularizer seperti *L1 Regularizer*, *L2 Regularizer*, dan sebagainya

Google Colab adalah salah satu produk buatan Google berbasis *cloud* yang bisa digunakan bersama secara *real-time* pada browser (seperti Google Docs dan Google Sheets). Google Colab bisa digunakan secara gratis maupun berbayar, tentunya dengan fitur lebih untuk yang berbayar. Kegunaan Google Colab ini adalah ditujukan bagi pengguna yang perlu mengerjakan *coding* bersama rekan secara *real-time* dan lebih berfokus pada pengguna yang tidak memiliki komputer/laptop dengan kapasitas yang mumpuni untuk membuat program atau dengan kata lain Google Colab meminjam kapasitas komputernya. Menurut Adam (2019), manfaat dari Google Colab adalah sebagai berikut:

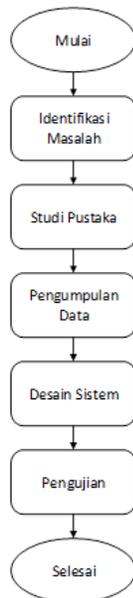
1. *Free GPU*: Google Colab memudahkan untuk menjalankan program pada komputer dengan spesifikasi tinggi (GPU Tesla, RAM hingga 12GB, Disk hingga 300GB yang masih bisa tersambung dengan Google Drive, akses internet cepat untuk *download file* besar) dan *running* dalam waktu yang lama (untuk versi gratisnya, Google Colab mengizinkan *running* program hingga 12 jam)
2. *Colaborate*: Google Colab juga memudahkan kolaborasi

dengan rekan dengan cara membagi *coding* secara *online* (mirip Google Doc). Dengan bergitu akan lebih mudah bereksperimen secara bersamaan, atau sekadar menggunakan fitur ini untuk mempelajari *coding* orang lain yang telah rapi

3. Mudah berintegrasi: Google Colab terbilang sangat fleksibel dalam hal integrasi. Google Colab dapat dengan mudah dihubungkan dengan Jupyter Notebook di komputer pribadi (*local runtime*), menghubungkan dengan Google Drive, atau dengan Github
4. Fleksibel: Google Colab memungkinkan *running* program via *Smartphone*. Hal ini bisa terjadi karena pada Google Colab hanya perlu *running* di browser. Dengan begitu pengawasan *training* atau proses jalannya *coding* bisa dipantau dengan mudah karena bisa diawasi dengan menggunakan *Smartphone* saja

### Analisis dan Perancangan Sistem

Alur penelitian ini terdiri dari identifikasi masalah, studi pustaka, pengumpulan data, desain sistem, dan pengujian seperti pada *flowchart* berikut ini.



Gambar 12. Alur Penelitian

#### 1. Identifikasi Masalah

Identifikasi masalah yang ada yaitu *trading* Forex tidak mudah, diperlukan analisis data menggunakan teknik yang tepat untuk membantu *trader* dalam melakukan transaksi pada Forex. Pada penelitian ini, menguji salah satu metode yang cocok untuk data *timeseries* yaitu *Long Shor-Term Memory*. LSTM diuji pada prediksi harga untuk mata uang EUR/USD dan GBP/USD.

#### 2. Studi Pustaka

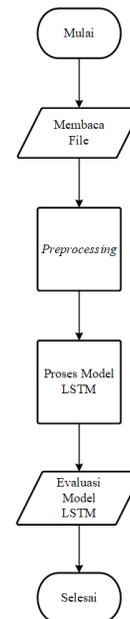
Identifikasi masalah yang ada yaitu *trading* Forex tidak mudah, diperlukan analisis data menggunakan teknik yang tepat untuk

membantu *trader* dalam melakukan transaksi pada Forex. Pada penelitian ini, menguji salah satu metode yang cocok untuk data *timeseries* yaitu *Long Shor-Term Memory*. LSTM diuji pada prediksi harga untuk mata uang EUR/USD dan GBP/USD.

#### 3. Pengumpulan Dataset

Pada tahap studi pustaka, peneliti menelusuri semua kebutuhan yang dibutuhkan sistem. Penelitian terdahulu, buku, dan *website* yang menunjang penelitian ini akan dijadikan referensi dalam pemecahan masalah. Pada penelitian ini menggunakan metode *Long Short-Term Memory* yang adalah salah satu varian dari *Recurrent Neural Network* yang cocok untuk digunakan pada data *timeseries*. Tahap studi pustaka ini mempelajari dasar teori dari LSTM dan alat pendukung pada penelitian ini.

#### 4. Desain Sistem



Gambar 13. Flowchart Desain Sistem

Desain sistem yang digunakan adalah *input berupa file* CSV yang berisi *dataset* dengan atribut *date*, *close*, *open*, *high*, *low*, dan *change*. Tahap selanjutnya adalah *preprocessing* yaitu mengurutkan data dari 1 Januari 2019 hingga 31 Desember 2020. Lalu menghilangkan atribut yang tidak digunakan yaitu menghilangkan atribut *date* dan *change*. Pada percobaan yang menggunakan 1 *input*, atribut *open*, *high*, dan *low* tidak digunakan, hanya data *close* saja yang digunakan. Pada percobaan dengan 3 *input*, data yang digunakan adalah *open* dan *close* pada 2 hari dan pada percobaan dengan 5 *input*, data yang digunakan *open* dan *close* pada 3 hari.

Setelah itu, *dataset* akan distandarisasi dan dibagi menjadi 2 bagian yaitu *train data* dan *test data* dengan perbandingan 80:20 yaitu *train data* berjumlah 418 data dan *test data* berjumlah 105 data. *Train data* akan digunakan sebagai pelatihan model LSTM, sedangkan *test data* akan digunakan

untuk menguji model LSTM yang telah dibuat. Pembagian *dataset* ini adalah dengan menggunakan data 1 Januari 2019 hingga 6 Agustus 2020 sebagai *train data* dan menggunakan data 7 Agustus 2020 hingga 31 Desember 2020 sebagai *test data*. Tahap selanjutnya adalah pembuatan model LSTM. Pada tahap ini dilakukan percobaan terhadap beberapa *input* yaitu *input* berjumlah 1, 3, dan 5 dan *output*-nya berjumlah 1 yaitu harga *close* hari ini. Pada *input* berjumlah 1 data yang digunakan adalah harga *close* kemarin (1). *Input* berjumlah 3 data yang digunakan adalah harga *open* pada 2 hari yaitu kemarin dan hari ini (2), dan *close* kemarin (1). Sedangkan untuk *input* berjumlah 5, data yang digunakan adalah data harga *open* pada 3 hari yaitu 2 hari lalu, kemarin, dan hari ini (3) dan *close* pada 2 hari yaitu 2 hari lalu dan kemarin (2). Berikut ini adalah arsitektur LSTM yang digunakan.

Tabel 2. Arsitektur LSTM

Layer	Output Shape
LSTM	(None, 10 atau 50)
Dense	(None, 1)

Arsitektur LSTM yang dibuat memiliki *input shape* 1x1 untuk percobaan dengan 1 *input*, 1x3 untuk percobaan dengan 3 *input*, dan 1x5 untuk percobaan dengan 5 *input*. Percobaan ini menggunakan 1 layer LSTM dengan jumlah *node* 10 dan 50, *optimizer* Adam dan Nadam, *activation function* menggunakan ReLu, dan *loss function* menggunakan MSE yang setelah model dibuat akan dievaluasi menggunakan MSE tersebut. Hasil *output* dari model LSTM ini adalah harga *close* hari ini. Arsitektur ini bisa berubah sesuai dengan percobaan yang dilakukan. Jumlah *epoch* sebanyak 200 dan menggunakan *early stopping*.

### 5. Pengujian Model

Pengujian model LSTM dilakukan dengan melihat hasil *Mean Squared Error* (MSE) yang didapatkan. MSE memberi informasi seberapa dekat garis regresi dengan sekumpulan titik. Jarak dari titik ke garis regresi (*error*) dikuadratkan agar menghindari adanya hasil negatif serta memberi bobot lebih pada perbedaan yang lebih besar. Berikut ini adalah rumus dari MSE.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2 \tag{1}$$

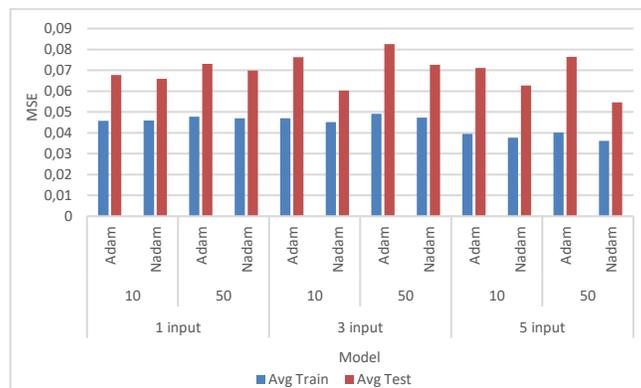
MSE bekerja dengan mencari rata-rata kesalahan (*error*) dari sekumpulan titik.  $y_i$  adalah *output* hasil prediksi dan  $\tilde{y}_i$  adalah *output* sesungguhnya. Semakin kecil hasil yang didapatkan, maka semakin baik model yang dibuat.

## Hasil dan Pembahasan

### 1. Hasil EUR/USD

Pada gambar 14 di bawah ini, dapat dilihat nilai rata-rata MSE terendah pada *train* 1 *input* adalah model dengan 1 layer LSTM

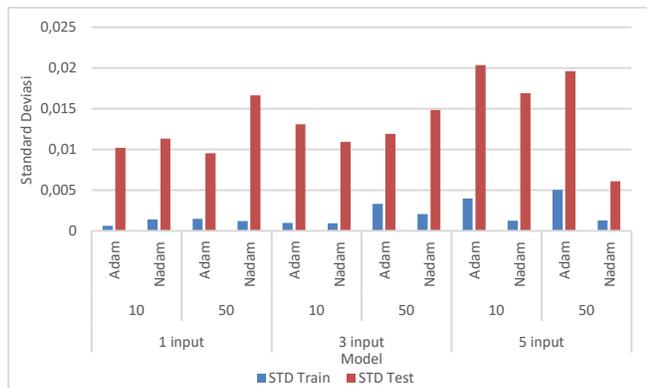
10 *node optimizer* Adam. Pada *test* 1 *input*, nilai rata-rata MSE terendah adalah model dengan 1 layer LSTM 10 *node optimizer* Nadam. Pada model *train* dan *test* dengan 3 *input*, model terbaik dengan rata-rata MSE terendah adalah model dengan 1 layer LSTM 10 *node optimizer* Nadam. Sedangkan pada model *train* 5 *input*, model terbaiknya ada pada model dengan 1 layer LSTM 10 *node optimizer* Nadam, dan model *test* 5 *input* terbaik pada model dengan 1 layer LSTM 50 *node optimizer* Nadam.



Gambar 14. Grafik Perbandingan Average Train dan Test EUR/USD

Dapat dilihat pada grafik tersebut (gambar 14), bahwa performa *optimizer* Nadam cenderung lebih baik bila dibandingkan dengan Adam, nilai MSE yang didapatkan Nadam lebih kecil daripada Adam. Hal ini kemungkinan dikarenakan *optimizer* Nadam diciptakan untuk memperbaiki kekurangan dan *optimizer* Adam. Sedangkan untuk model dengan 1 layer LSTM 10 *node* cenderung mendapatkan hasil MSE lebih baik jika dibandingkan dengan model dengan 1 layer LSTM 50 *node*. Model akan lebih baik jika menggunakan jumlah *node* yang lebih sedikit untuk menghindari *memory loss* atau *error* pada *memory cell* LSTM.

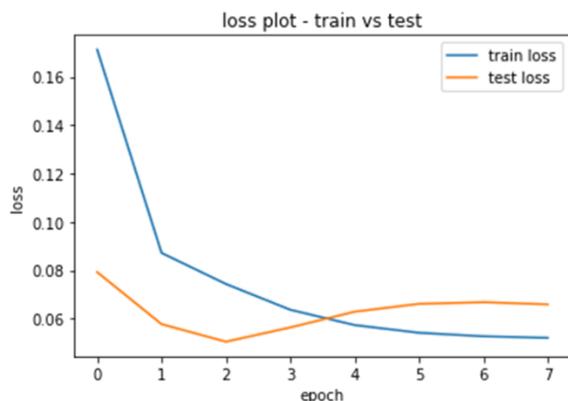
Gambar 15 adalah grafik perbandingan standard deviasi pada setiap model EUR/USD. Berdasarkan grafik tersebut, standard deviasi yang didapatkan tidak terlalu besar, sehingga data tersebut tidak memiliki persebaran yang bervariasi. Pada model *train* dengan 1 *input*, standard deviasi terkecil ada pada model dengan 1 layer LSTM 10 *node optimizer* Adam, dan standard deviasi terkecil model *test* pada model dengan 1 layer LSTM 50 *node optimizer* Adam. Model *train* maupun *test* 3 *input*, standard deviasi terkecil adalah model dengan 1 layer LSTM 50 *node optimizer* Nadam. Sedangkan untuk model 5 *input*, model *train* dengan standard deviasi terkecil adalah model dengan 1 layer LSTM 10 *node optimizer* Nadam dan model *test* dengan standard deviasi terkecil adalah model dengan 1 layer LSTM 50 *node optimizer* Nadam.



Gambar 15. Grafik Perbandingan Standard Deviasi Train dan Test EUR/USD

Model EUR/USD dengan 1 *input* terbaik adalah model dengan 1 layer LSTM 10 *node* dan menggunakan *optimizer* Nadam dengan nilai MSE terbaik 0,0535. Pada model 3 *input*, model terbaik adalah model dengan 1 layer LSTM 10 *node* dan menggunakan *optimizer* Nadam yang mendapatkan nilai MSE sebesar 0,0529. Sedangkan, pada model EUR/USD dengan 5 *input* terbaik adalah model dengan 1 layer LSTM 10 *node* dan menggunakan *optimizer* Nadam dengan nilai MSE terbaik 0,0469. Model terbaik ini yang digunakan pada *website* yang dibuat.

Gambar 16 di bawah ini merupakan *loss plot* dari model 5 *input* dengan 1 layer LSTM menggunakan *optimizer* Nadam. Dari grafik tersebut dapat dilihat bahwa model tersebut mengalami *over-fitting*. Grafik *train loss* terus mengalami penurunan, namun untuk *test loss*-nya sempat mengalami penurunan dan kembali naik.

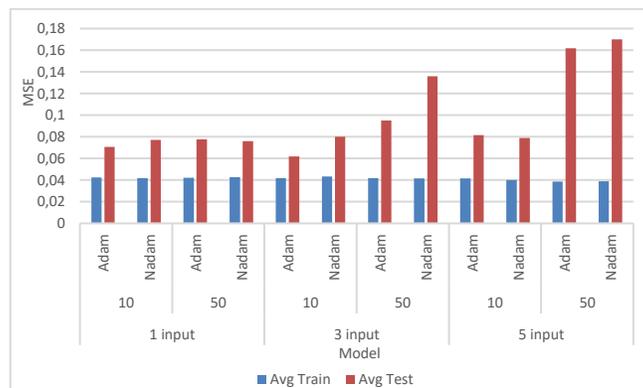


Gambar 16. Grafik Loss Plot EUR/USD

## 2. Hasil GBP/USD

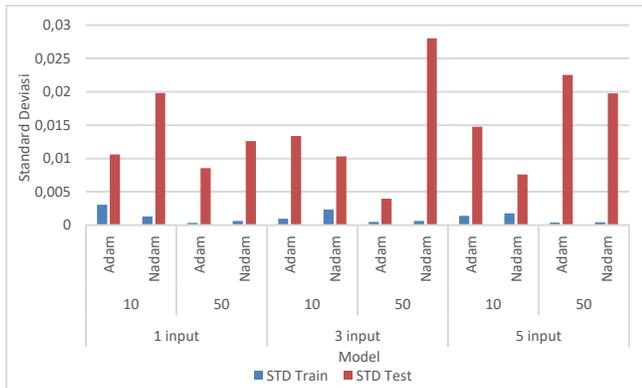
Gambar 17 di bawah ini merupakan grafik perbandingan *average train* dan *test* mata uang GBP/USD. Dapat dilihat nilai rata-rata MSE pada seluruh model *train* hampir sama. Rata-rata MSE pada model *test* lebih bervariasi. Nilai rata-rata MSE terendah pada model *train* 1 *input* adalah model dengan 1 layer LSTM 10 *node optimizer* Nadam. Pada *test* 1 *input*, nilai rata-rata MSE terendah adalah model dengan 1 layer LSTM 10 *node*

*optimizer* Adam. Pada model *train* 3 *input*, model terbaik dengan rata-rata MSE terendah adalah model dengan 1 layer LSTM 50 *node optimizer* Nadam dan model *test* dengan rata-rata MSE terendah ada pada model dengan 1 layer LSTM 10 *node optimizer* Adam. Sedangkan pada model *train* 5 *input*, model terbaiknya ada pada model dengan 1 layer LSTM 50 *node optimizer* Adam, dan model *test* 5 *input* terbaik pada model dengan 1 layer LSTM 10 *node optimizer* Nadam.



Gambar 17. Grafik Perhitungan Akurasi MSE Mata Uang GBP/USD

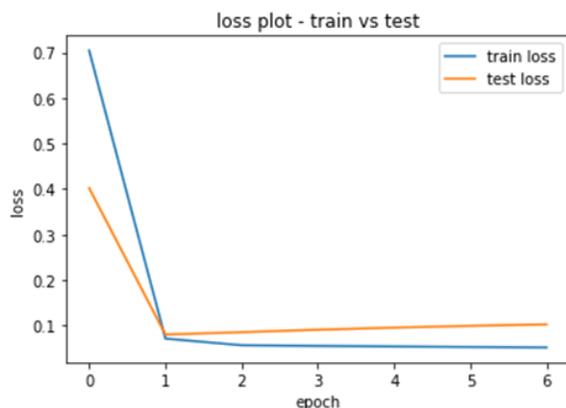
Pada grafik di atas (gambar 17), dapat dilihat bahwa performa *optimizer* Adam cenderung lebih baik bila dibandingkan dengan Nadam, nilai MSE yang didapatkan Adam lebih kecil daripada Nadam. Pada mata uang GBP/USD ini, *optimizer* Nadam tidak berhasil mengungguli *optimizer* sebelumnya yaitu Adam. Sedangkan untuk model dengan 1 layer LSTM 10 *node* cenderung mendapatkan hasil MSE lebih baik jika dibandingkan dengan model dengan 1 layer LSTM 50 *node*. Sama dengan mata uang EUR/USD, model akan lebih baik jika menggunakan jumlah *node* yang lebih sedikit untuk menghindari *memory loss* atau *error* pada *memory cell* LSTM. Gambar 18 adalah grafik perbandingan standard deviasi pada setiap model GBP/USD. Berdasarkan grafik tersebut, sama seperti grafik standard deviasi pada mata uang EUR/USD, standard deviasi yang didapatkan tidak terlalu besar, sehingga data tersebut tidak memiliki persebaran yang bervariasi. Pada model *train* dan *test* dengan 1 *input*, standard deviasi terkecil ada pada model dengan 1 layer LSTM 50 *node optimizer* Adam. Pada model *train* dan *test* 3 *input*, standard deviasi terkecil adalah model dengan 1 layer LSTM 50 *node optimizer* Adam. Sedangkan untuk model 5 *input*, model *train* dengan standard deviasi terkecil adalah model dengan 1 layer LSTM 50 *node optimizer* Adam dan model *test* dengan standard deviasi terkecil adalah model dengan 1 layer LSTM 10 *node optimizer* Nadam.



Gambar 18. Grafik Perbandingan Standard Deviasi Train dan Test GBP/USD

Model GBP/USD dengan 1 input terbaik adalah model dengan 1 layer LSTM 10 node dan menggunakan optimizer Nadam dengan nilai MSE terbaik 0,0543. Pada model 3 input, model terbaik adalah model dengan 1 layer LSTM 10 node dan menggunakan optimizer Adam yang mendapatkan nilai MSE sebesar 0,0520. Sedangkan, pada model EUR/USD dengan 5 input terbaik adalah model dengan 1 layer LSTM 10 node dan menggunakan optimizer Adam dengan nilai MSE terbaik 0,0631. Model terbaik ini yang digunakan pada website yang dibuat.

Gambar 19 di bawah ini merupakan loss plot dari model 3 input dengan 1 layer LSTM menggunakan optimizer Adam. Dari grafik tersebut dapat dilihat bahwa model tersebut mengalami over-fitting. Grafik train loss terus mengalami penurunan, namun untuk test loss-nya sempat mengalami penurunan dan kembali naik.



Gambar 19. Grafik Loss Plot GBP/USD

### 3. Tampilan Website

Tampilan “Home” dari website yang dibuat adalah seperti pada gambar 20. Pada bagian ini adalah halaman awal yang terdapat 2 button yaitu EUR/USD dan GBP/USD. Jika ingin memprediksi mata uang Euro terhadap Dolar Amerika, maka tekan button “EUR/USD” dan jika ingin memprediksi mata uang Pound Sterling terhadap Dolar Amerika, tekan button “GBP/USD”.



Gambar 20. Tampilan Home

Tampilan EUR/USD akan muncul jika pada halaman “Home” menekan button “EUR/USD”. Gambar 21 merupakan tampilan awal EUR/USD. Terdapat 4 button yaitu “Model 1 Input”, “Model 3 Input”, “Model 5 Input” dan “Home”.



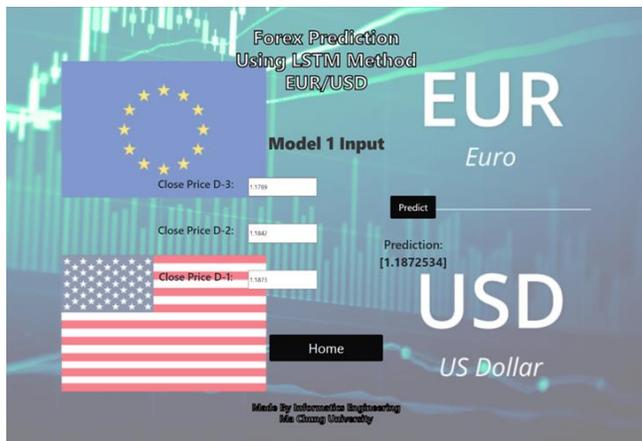
Gambar 21. Tampilan EUR/USD

Gambar 22 merupakan tampilan jika menekan button “Model 1 Input”.



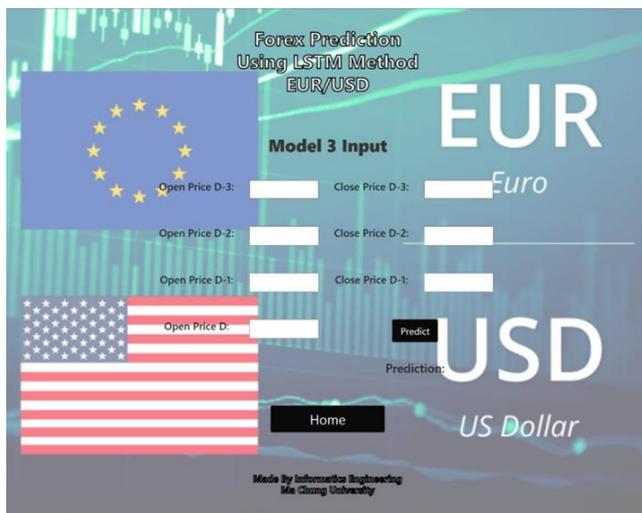
Gambar 22. Tampilan EUR/USD 1 Input

Gambar 23 merupakan contoh tampilan jika sudah memberi input dan menekan tombol “Predict”.



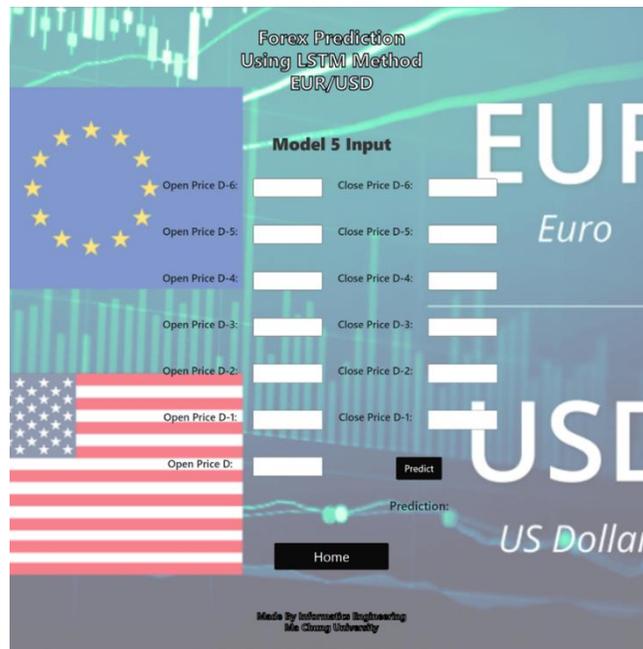
Gambar 23. Tampilan EUR/USD 1 Input dengan Prediksi

Gambar 24 merupakan tampilan jika menekan button “Model 3 Input”.



Gambar 24. Tampilan EUR/USD 3 Input

Gambar 25 merupakan tampilan jika menekan button “Model 5 Input”.



Gambar 25. Tampilan EUR/USD 5 Input

Tampilan untuk mata uang GBP/USD tidak jauh berbeda dengan EUR/USD.

#### 4. Pengujian Model Pada Simulasi

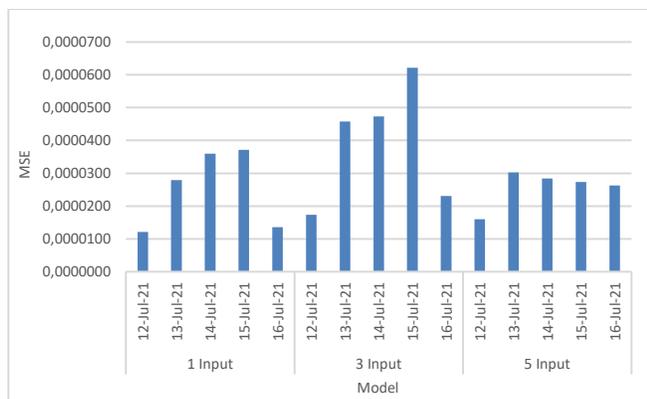
Model yang digunakan pada website diuji coba untuk data pada 5 hari yaitu 12 Juli 2021 hingga 16 Juli 2021 dan dihitung akurasi MSE-nya. Lalu dilakukan simulasi *open position* 1 lot pada harga *open* dan *take profit* (TP) pada harga prediksi. Pada saat *closing candle* 1 harian tersebut, dilihat apakah harga menyentuh harga TP dan dilakukan perhitungan akurasinya. Tabel 3 berikut ini contoh cara perhitungan akurasi MSE pada model 1 *input* mata uang EUR/USD tanggal 12 Juli 2021. Data *close* dari 3 hari sebelumnya hingga 1 hari sebelumnya (7 Juli 2021 hingga 9 Juli 2021) dimasukkan ke dalam website, lalu hasil prediksi dan nilai asli dihitung selisihnya dan dikuadratkan sehingga didapatkan *squared error*. Hasil akurasi MSE didapatkan dari rata-rata *squared error* tersebut, sehingga tanggal 12 Juli 2021 mendapatkan hasil akurasi MSE sebesar 0,0000121.

Tabel 3 Contoh Perhitungan Akurasi MSE

Day	Actual	Prediction	Squared Error
D-3	1,1789	-	-
D-2	1,1842	1,1791	0,0000255
D-1	1,1873	1,1843	0,0000089
D	1,1859	1,1873	0,0000018
		<b>MSE</b>	<b>0,0000121</b>

a. EUR/USD

Grafik gambar 26 adalah hasil perhitungan akurasi MSE pada mata uang EUR/USD. Pada model 1 *input*, rata-rata perhitungan akurasi MSE yang didapatkan adalah 0,0000253. Pada model EUR/USD 3 *input*, rata-rata akurasi MSE yang didapatkan adalah 0,0000391. Sedangkan hasil rata-rata akurasi MSE yang didapatkan model EUR/USD 5 *input* adalah 0,0000256. Akurasi MSE yang didapatkan model dengan 1 *input* dan 5 *input* lebih baik jika dibandingkan dengan model dengan 3 *input*.



Gambar 26. Grafik Perhitungan Akurasi MSE Mata Uang EUR/USD

Tabel 4 di bawah ini adalah tabel simulasi yang dilakukan selama 5 hari pada mata uang EUR/USD. Hasil yang didapatkan adalah untuk model 1 *input*, prediksi yang dilakukan selama 5 hari semuanya berhasil menyentuh TP, namun pada tanggal 15 Juli 2021 prediksi harga sama dengan harga *open* dan tidak mendapatkan profit, sehingga akurasinya sebesar 80%. Pada model 3 *input*, pada tanggal 14 Juli 2021, harga prediksi tidak berhasil menyentuh TP, namun untuk hari lainnya semua berhasil menyentuh TP, sehingga akurasinya sebesar 80%. Sedangkan untuk model 5 *input*, sama seperti model 1 *input*, semuanya berhasil menyentuh TP, namun pada tanggal 16 Juli 2021 prediksi harga sama dengan harga *open* dan tidak mendapatkan profit, sehingga akurasinya sebesar 80%. Dari semua model tersebut, jika dirata-rata maka akurasi untuk percobaan 5 hari mata uang EUR/USD adalah sebesar 80%.

Pada percobaan 5 hari ini, sebesar 7 *pip* yang dihasilkan oleh prediksi model 1 *input*, 15 *pip* dihasilkan oleh prediksi model 3 *input*, dan 23 *pip* yang dihasilkan oleh prediksi model 5 *input*. Rata-rata yang dihasilkan adalah 15 *pip*. Pada sisi *trader* tentu jumlah ini tergolong kecil jika tidak membuka lot yang besar dan tidak cocok bila menggunakan broker yang memiliki sistem menggunakan *spread* tinggi.

Tabel 4. Take Profit Mata Uang EUR/USD

Date	In put	Open	High	Low	Close	Predi ction	Take Profit	Profit (Pip)
12-Jul	1	1,187 2	1,188 1	1,183 6	1,185 9	1,187 3	YES	1

	3	1,187 2	1,188 1	1,183 6	1,185 9	1,187 4	YES	2
	5	1,187 2	1,188 1	1,183 6	1,185 9	1,186 6	YES	6
13-Jul	1	1,186 2	1,187 6	1,177 2	1,177 4	1,185 9	YES	3
	3	1,186 2	1,187 6	1,177 2	1,177 4	1,186 1	YES	1
	5	1,186 2	1,187 6	1,177 2	1,177 4	1,186 4	YES	2
14-Jul	1	1,177 6	1,183 9	1,177 2	1,183 5	1,177 8	YES	2
	3	1,177 6	1,183 9	1,177 2	1,183 5	1,175 4	NO	-
	5	1,177 6	1,183 9	1,177 2	1,183 5	1,177 4	YES	2
15-Jul	1	1,183 7	1,185 2	1,179 6	1,181 2	1,183 7	YES	0
	3	1,183 7	1,185 2	1,179 6	1,181 2	1,182 7	YES	10
	5	1,183 7	1,185 2	1,179 6	1,181 2	1,182 4	YES	13
16-Jul	1	1,181 4	1,182 3	1,179 1	1,180 5	1,181 3	YES	1
	3	1,181 4	1,182 3	1,179 1	1,180 5	1,181 6	YES	2
	5	1,181 4	1,182 3	1,179 1	1,180 5	1,181 4	YES	0

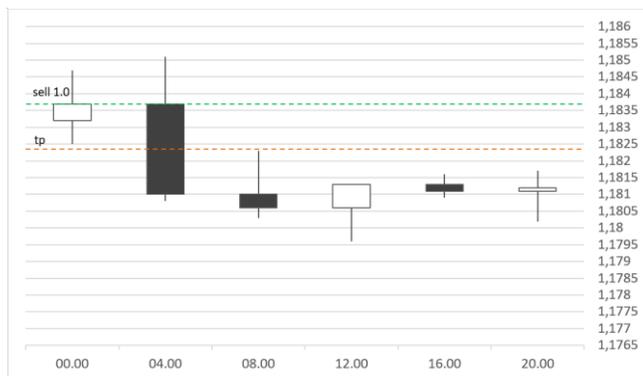
Tabel 5 merupakan hasil simulasi percobaan pada salah 1 broker yang terdaftar di BAPPEPTI dengan anggapan komisi \$5/lot dan *spread* rata-rata 1 *pip* pada saat *open position* dan *take profit*, sehingga *spread* 1x2 yaitu 2 *pip*.

Tabel 5. Simulasi Percobaan Mata Uang EUR/USD Pada Broker

Date	Input	Open	Prediction	Profit (\$)
12-Jul	1	1,1872	1,1873	-15
	3	1,1872	1,1874	-5
	5	1,1872	1,1866	35
13-Jul	1	1,1862	1,1859	5
	3	1,1862	1,1861	-15
	5	1,1862	1,1864	-5
14-Jul	1	1,1776	1,1778	-5
	3	1,1776	1,1754	-
	5	1,1776	1,1774	-5

15-Jul	1	1,1837	1,1837	-
	3	1,1837	1,1827	75
	5	1,1837	1,1824	105
16-Jul	1	1,1814	1,1813	-15
	3	1,1814	1,1816	-5
	5	1,1814	1,1814	-

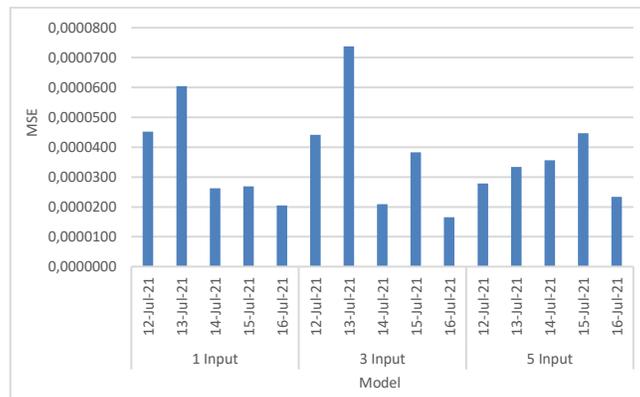
Pada model 1 input mendapatkan loss sebesar \$30, pada model 3 input sebesar \$50 dan model 5 input mendapat profit sebesar \$130. Rata-rata profit yang didapatkan adalah \$50. Pada mata uang EUR/USD ini model dengan 5 input mendapatkan profit terbesar jika dibandingkan dengan model 1 input dan 3 input. Gambar 27 di bawah ini merupakan contoh dari open position sell pada tanggal 15 Juli 2021 dengan menggunakan harga open (1,1824) sebagai open position sell dan menggunakan harga prediksi model 5 input (1,1824) sebagai take profit.



Gambar 27. Simulasi Open Position dan Take Profit Pada Candle EUR/USD

b. GBP/USD

Grafik gambar 28 adalah hasil perhitungan akurasi MSE pada mata uang GBPP/USD. Hasil rata-rata akurasi MSE yang didapatkan dari model GBP/USD 1 input adalah 0,0000359. Pada model GBP/USD 3 input hasil rata-rata akurasi MSE yang didapatkan adalah 0,0000387. Sedangkan hasil rata-rata akurasi MSE yang didapatkan dari model GBP/USD 5 input adalah 0,0000330. Hasil akurasi MSE pada model 5 input mendapatkan hasil lebih baik jika dibandingkan dengan model 1 input dan 3 input.



Gambar 28. Grafik Perhitungan Akurasi MSE Mata Uang GBP/USD

Tabel 6 adalah tabel simulasi yang dilakukan selama 5 hari pada mata uang GBP/USD. Dapat dilihat pada tabel tersebut bahwa hasil yang didapatkan pada model 1 input prediksi yang dilakukan selama 5 hari semuanya berhasil menyentuh TP, namun pada tanggal 15 Juli 2021 prediksi harga sama dengan harga open dan tidak mendapatkan profit, sehingga akurasinya sebesar 80%. Pada model 3 input dan 5 input pada percobaan 5 hari semuanya berhasil menyentuh TP, sehingga akurasinya sebesar 100%. Dari semua model tersebut, jika dirata-rata maka akurasi untuk percobaan 5 hari mata uang GBP/USD adalah sebesar 93,33%.

Pada percobaan 5 hari ini, sebesar 17 pip yang dihasilkan oleh prediksi model 1 input, 89 pip dihasilkan oleh prediksi model 3 input, dan 64 pip yang dihasilkan oleh prediksi model 5 input. Rata-rata yang dihasilkan adalah 56,67 pip. Pada sisi trader jumlah ini tergolong kecil jika tidak membuka lot yang besar dan tidak cocok bila menggunakan broker yang memiliki sistem menggunakan spread tinggi.

Tabel 6. Take Profit Mata Uang GBP/USD

Date	In put	Open	High	Low	Clos e	Predi ction	Take Profit	Profit (Pip)
12-Jul	1	1,3902	1,3911	1,3838	1,3880	1,3895	YES	7
	3	1,3902	1,3911	1,3838	1,3880	1,3841	YES	61
	5	1,3902	1,3911	1,3838	1,3880	1,3910	YES	8
13-Jul	1	1,3885	1,3907	1,3800	1,3808	1,3881	YES	4
	3	1,3885	1,3907	1,3800	1,3808	1,3877	YES	8
	5	1,3885	1,3907	1,3800	1,3808	1,3864	YES	21
14-Jul	1	1,3814	1,3893	1,3801	1,3856	1,3810	YES	4
	3	1,3814	1,3893	1,3801	1,3856	1,3824	YES	10
	5	1,3814	1,3893	1,3801	1,3856	1,3823	YES	9

15-Jul	1	1,3858	1,3900	1,3805	1,3828	1,3858	YES	0
	3	1,3858	1,3900	1,3805	1,3828	1,3853	YES	5
	5	1,3858	1,3900	1,3805	1,3828	1,3873	YES	15
16-Jul	1	1,383	1,3863	1,3759	1,3772	1,3828	YES	2
	3	1,383	1,3863	1,3759	1,3772	1,3835	YES	5
	5	1,383	1,3863	1,3759	1,3772	1,3841	YES	11

Tabel 7 merupakan hasil simulasi percobaan pada salah 1 broker yang terdaftar di BAPPEPTI dengan anggapan komisi \$5/lot dan spread rata-rata 1 pip pada saat open position dan take profit, sehingga spread 1x2 yaitu 2 pip.

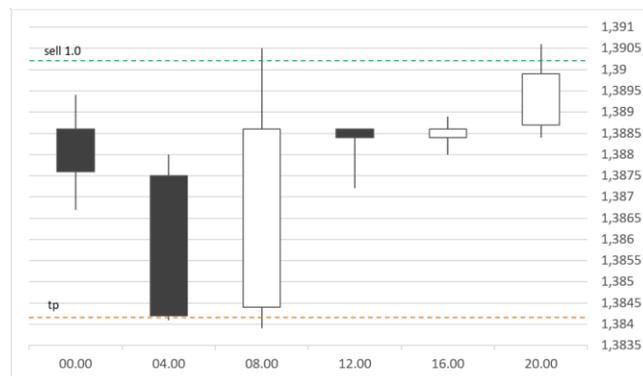
Tabel 7. Simulasi Percobaan Mata Uang GBP/USD Pada Broker

Date	Input	Open	Prediction	Profit (\$)
12-Jul	1	1,3902	1,3895	45
	3	1,3902	1,3841	585
	5	1,3902	1,3910	55
13-Jul	1	1,3885	1,3881	15
	3	1,3885	1,3877	55
	5	1,3885	1,3864	185
14-Jul	1	1,3814	1,3810	15
	3	1,3814	1,3824	75
	5	1,3814	1,3823	65
15-Jul	1	1,3858	1,3858	-
	3	1,3858	1,3853	25
	5	1,3858	1,3873	125
16-Jul	1	1,383	1,3828	-5
	3	1,383	1,3835	25
	5	1,383	1,3841	85

Pada model 1 input mendapatkan profit sebesar \$70, pada model 3 input sebesar \$765 dan model 5 input mendapat profit sebesar \$515. Rata-rata profit yang didapatkan adalah \$450.

Pada mata uang GBP/USD ini model dengan 3 input mendapatkan profit terbesar jika dibandingkan dengan model 1 input dan 5 input.

Gambar 29 di bawah ini merupakan contoh dari open position sell pada tanggal 12 Juli 2021 dengan menggunakan harga open (1,3902) sebagai open position sell dan menggunakan harga prediksi model 3 input (1,3841) sebagai take profit.



Gambar 29 Simulasi Open Position dan Take Profit Pada Candle GBP/USD

### Simpulan dan Saran

1. Simpulan
 

Berdasarkan hasil percobaan yang telah dilakukan, maka didapatkan kesimpulan sebagai berikut.

  - a. Model LSTM berhasil digunakan dalam prediksi harga Forex mata uang EUR/USD dan GBP/USD.
  - b. Model EUR/USD dengan 1 input mendapatkan nilai loss MSE terendah yaitu 0,0535 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Nadam. Pada model 3 input mendapatkan nilai loss MSE terendah yaitu 0,0529 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Nadam. Sedangkan, model 5 input mendapatkan nilai loss MSE terendah yaitu 0,0469 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Nadam.
  - c. Model GBP/USD dengan 1 input mendapatkan nilai loss MSE terendah yaitu 0,0543 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Nadam. Pada model 3 input mendapatkan nilai loss MSE terendah yaitu 0,0520 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Adam. Sedangkan, model 5 input mendapatkan nilai loss MSE terendah yaitu 0,0631 dengan model 1 layer LSTM 10 node dan menggunakan optimizer Adam.
  - d. Pada perhitungan akurasi MSE pada percobaan 5 hari, hasil rata-rata yang didapatkan adalah model EUR/USD 1 input sebesar 0,0000253, 3 input sebesar 0,0000368, dan 5 input sebesar 0,0000249. Sedangkan pada model GBP/USD 1 input sebesar 0,0000439, 3 input sebesar 0,0000463, dan 5 input sebesar 0,0000323.
  - e. Pada percobaan simulasi 5 hari dengan menggunakan harga open sebagai open position dan harga prediksi sebagai take profit. Pada mata uang EUR/USD maupun GBP/USD

mendapatkan hasil yang baik yaitu dengan akurasi 80% untuk mata uang EUR/USD dan 93,33% untuk mata uang GBP/USD. Rata-rata profit yang didapatkan mata uang EUR/USD adalah \$50 dan rata-rata profit yang didapatkan mata uang GBP/USD adalah \$450.

- f. Rekomendasi yang diberikan adalah pada mata uang EUR/USD menggunakan model 5 *input* dan mata uang GBP/USD menggunakan model 3 *input*. Pertimbangannya adalah hasil *loss* MSE, akurasi MSE, dan jumlah *profit* yang didapatkan semuanya adalah yang terbaik jika dibandingkan dengan model dengan jumlah *input* lainnya.
- g. Aplikasi *website* dapat memberikan prediksi harga Forex mata uang EUR/USD dan GBP/USD menggunakan model dengan nilai MSE terbaik.

## 2. Saran

Berdasarkan simpulan yang didapatkan, beberapa saran yang dapat dilakukan sebagai berikut.

- a. Perlu adanya penambahan dataset harga dalam rentang waktu yang lebih panjang.
- b. Diperlukan pencarian metode preprocessing dan arsitektur model LSTM yang lebih tepat untuk menghilangkan overfitting, selain itu juga bisa ditambahkan regularisasi.
- c. Melakukan prediksi harga dengan metode GRU (varian lain dari RNN).

## Daftar Pustaka

- William, H., A. Hidayatno, dan A. A. Zahra. (2014). Aplikasi Jaringan Saraf Tiruan Perambatan Balik Untuk Prakiraan Valuta GBP/USD Dalam Forex Trading. *Jurnal Transient*, 3(4): 493-500.
- Dautel A. J., Hardle, W. K., Lessmann, S., Seow, H. V. (2020). Forex Exchange Rate Forecasting using Deep Recurrent Neural Networks. *Journal Digital Finance*, 2:69-96.
- Yildirim, D. C., Toroslu, I. H., Fiore, U. (2021). Forecasting Directional Movement of Forex Data using LSTM with Technical and Macroeconomic Indicators. *Journal Springer Open*, 7(1): 1-36.
- Sihombing, M, (3 Maret 2021). Kamus Ekonomi: Apa Arti Forex? URL <https://ekonomi.bisnis.com/read/20130729/9/153630/kamus-ekonomi-apa-arti-forex>.
- Dahria, M. (2008). Kecerdasan Buatan (Artificial Intelligence). *Jurnal SAINTIKOM*, 5(2): 185-196.
- Fachrizal, R. (31 Agustus 2020). Mengenal Beberapa Perpustakaan Machine Learning Berbasis Python. URL <https://infokomputer.grid.id/read/121744301/mengenal-beberapa-perpustakaan-machine-learning-berbasis-python?page=all>.
- Peryanto, A., Yudhana, A., and Umar, R. (2019). Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network. *Jurnal Format*, 8(2): 138-147.
- Widiputra, H. D. (2 September 2020). Artificial Neural Network, Perbanas Institute. URL <https://dosen.perbanas.id/artificial-neural-network>.
- Mutmainnah, I. (25 Februari 2021). Mengenal Pandas Dalam Python. URL <https://medium.com/@16611092/mengenal-pandas-dalam-python-cc66d0c5ea40>.
- Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd edition*. O'Reilly Media: Sebastopol.
- Purwanto. (31 Agustus 2020). 3 Library Python Terbaik Untuk Data Science. URL <https://belajarpython.com/2018/09/3-library-python-terbaik-untuk-data-science.html>.
- Rohman, Y. A. (31 Agustus 2020). 3 Pengenalan NumPy, Pandas, Matplotlib. URL <https://medium.com/@yasirabd/pengenalan-numpy-pandas-matplotlib-b90bafd36c0>.
- Gifari, J. (25 Februari 2021). Belajar Data Science : Apa yang dimaksud dengan Tensorflow dan Bagaimana Penggunaannya? URL <https://www.dqlab.id/belajar-data-science-pahami-tensflow>.
- Adam, R. (1 September 2020). Mengenal Google Colab, *Structilmy*. URL <https://structilmy.com/2019/05/mengenal-google-colab/#:~:text=Seperti%20Google%20Drive%20Google%20Doc,akses%20komputer%20dengan%20spek%20tinggi>.