

PREDIKSI VALUTA ASING MENGGUNAKAN LSTM YANG DIOPTIMALKAN DENGAN ALGORITMA GENETIK

PROPOSAL TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Komputer
Program Studi Informatika



Diajukan oleh:
Daniel Budi Prasetyo
205314145

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2023**

PERSETUJUAN PEMBIMBING

PROPOSAL TUGAS AKHIR

PREDIKSI VALUTA ASING MENGGUNAKAN LSTM YANG DIOPTIMALKAN DENGAN ALGORITMA GENETIK

Disusun oleh:

Daniel Budi Prasetyo

205314145

Dosen Pembimbing,

(Cyprianus Kuntoro Adi, S.J. M.A., M.Sc., Ph.D.)

16 Juni 2023

DAFTAR ISI

PERSETUJUAN PEMBIMBING	i
DAFTAR ISI	ii
DAFTAR TABEL	iv
DAFTAR GAMBAR	v
ABSTRAK	vi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA	4
2.1. Tinjauan Pustaka	4
2.2. Landasan Teori	7
2.2.1. Valuta Asing	7
2.2.2. Preprocessing	7
2.2.3. Recurrent Neural Network (RNN)	9
2.2.4. Long Short Term Memory (LSTM).....	11
2.2.5. Algoritma Genetik.....	17
2.2.6. Evaluasi Model.....	20
BAB III METODE PENELITIAN	23
3.1. Deskripsi Data	23
3.2. Preprocessing.....	24
3.2.1. Normalisasi Data.....	24
3.2.2. Sliding Window	24
3.2.3. Split Data.....	24
3.3. Implementasi Model.....	25
3.3.1. Base LSTM	25
3.3.2. GA-LSTM.....	26
3.4. Evaluasi Model.....	28
3.5. Desain User Interface	28
3.6. Kebutuhan Hardware dan Software.....	29

3.7. Rancangan Skenario Pengujian	30
JADWAL KEGIATAN	31
DAFTAR PUSTAKA	32

DAFTAR TABEL

Tabel 2.1 Review Literatur.....	4
Tabel 3.1 Contoh Data Mentah	23
Tabel 3.2 Contoh Data Normalisasi	24
Tabel 3.3 Skenario Pengujian.....	30

DAFTAR GAMBAR

Gambar 2.1 Proses Sliding Window	9
Gambar 2.2 RNN memiliki loop	10
Gambar 2.3 Representasi langkah dari RNN	10
Gambar 2.4 RNN menggunakan keadaan jaringan sebelumnya.....	11
Gambar 2.5 Arsitektur LSTM	12
Gambar 2.6 Forget Gate	13
Gambar 2.7 Input Gate	14
Gambar 2.8 Cell State	15
Gambar 2.9 Output Gate	16
Gambar 2.10 Langkah Algoritma Genetik	17
Gambar 2.11 Tournament Selection	18
Gambar 2.12 Single-point Crossover	19
Gambar 2.13 Swap Mutation	20
Gambar 3.1 Langkah Penelitian	23
Gambar 3.2 Langkah Preprocessing	24
Gambar 3.3 Arsitektur LSTM	25
Gambar 3.4 Langkah GA-LSTM	26
Gambar 3.5 Rancangan Desain GUI.....	28

ABSTRAK

Foreign exchange (Forex) adalah salah satu pasar keuangan terbesar di dunia, dengan lebih dari \$5,1 triliun diperdagangkan setiap hari. Pada penelitian ini, *Long Short-Term Memory* (LSTM) dan *Genetic Algorithm Long Short-Term Memory* (GA-LSTM) digunakan untuk memprediksi bagaimana pola harga dari USD, EUR, dan JPY. Data diambil dari website Google Finance dalam kurun waktu 3 tahun dengan total data sekitar 1247 data.

Penelitian ini melakukan normalisasi data Forex menggunakan normalisasi *min-max*. Hasil normalisasi akan di-*reshape* sesuai dengan *sliding window*. Setelah data di-*reshape* data akan dibagi menjadi 2 bagian yaitu data *training* dan *test*. Data *training* dan *test* akan digunakan untuk melatih model.

Kata kunci : *Foreign exchange*, *Long Short-Term Memory*, Algoritma Genetik

BAB I

PENDAHULUAN

1.1. Latar Belakang

Foreign exchange (Forex) adalah salah satu pasar keuangan terbesar di dunia, dengan lebih dari \$5,1 triliun diperdagangkan setiap hari. Karena kompleksitas dan volatilitasnya, prediksi harga menjadi sulit [1]. Terutama, di negara berkembang seperti Indonesia, yang sangat penting untuk mendukung pembangunan ekonomi yang berkelanjutan dan meningkatkan kesejahteraan rakyat. Ketidakstabilan nilai tukar dapat menyurutkan minat investor untuk berinvestasi, yang dapat menyebabkan kemunduran dalam pembangunan di Indonesia. Sebab, selama ini peran investor asing sangat besar dalam pertumbuhan ekonomi [2].

Deep Learning telah mencapai kesuksesan besar di bidang *image recognition*, *natural language processing*, *speech recognition*, *video processing*, dan lain – lain. Oleh karena itu, penerapan algoritma *Deep Learning* dalam prediksi nilai tukar juga mendapat perhatian luas [3, 4, 5]. Peneliti keuangan di seluruh dunia telah mempelajari dan menganalisis perubahan di pasar saham dan Forex. Penerapan kecerdasan buatan yang meluas telah menyebabkan peningkatan jumlah investor yang menggunakan model *Deep Learning* untuk memprediksi dan mempelajari harga saham dan Forex. Telah terbukti bahwa fluktuasi harga saham dan Forex dapat diprediksi [4].

Berdasarkan salah satu literatur yang peneliti baca, model LSTM lebih baik dibandingkan dengan model RNN. Dimana model LSTM memiliki Root Mean Square Error (RMSE) dan Mean Absolute Error (MAE) yang lebih kecil dibandingkan dengan model RNN [6]. Dengan literatur di atas sebagai dasar, peneliti ingin mengambil model LSTM tersebut sebagai bahan penelitian untuk memprediksi harga valuta asing dalam 3 – 5 tahun terakhir. Selain itu, peneliti juga akan menggunakan Algoritma Genetik untuk mengoptimasi model LSTM, yang diharapkan akan menurunkan *error* atau kesalahan dari model awal.

1.2. Rumusan Masalah

1. Bagaimana tingkat evaluasi matriks menggunakan LSTM dan GA-LSTM untuk harga mata uang asing USD, EUR, dan JPY?
2. Apakah dengan optimasi parameter menggunakan Algoritma Genetik dapat menurunkan *error* atau kesalahan pada sebuah model?

1.3. Batasan Masalah

1. Data yang digunakan adalah nilai tukar untuk USD/IDR, EUR/IDR, dan JPY/IDR dengan rentang waktu 3 tahun terakhir.
2. Arsitektur model *Deep Learning* yang digunakan adalah LSTM.
3. Algoritma optimisasi yang digunakan adalah Algoritma Genetik.

1.4. Tujuan Penelitian

1. Untuk mengetahui perbandingan antara model awal dengan model yang telah dioptimalkan.
2. Untuk mengetahui apakah Algoritma Genetik berpengaruh terhadap *error* atau kesalahan dari sebuah model.

1.5. Manfaat Penelitian

1. Meningkatkan pemahaman tentang prediksi nilai tukar mata uang asing.
2. Memperluas pengetahuan dalam bidang kecerdasan buatan dan keuangan.
3. Membantu pengambilan keputusan yang lebih baik di pasar forex.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Tabel 2.1 Review Literatur

Peneliti	Judul	Model	Hasil
Lina Ni, Yujie Li, Xiao Wang, Jinquan Zhang, Jiguo Yu, Chengming Qi	Forecasting of Forex Time Series Data Based on Deep Learning (2019)	C-RNN, LSTM, CNN	Hasil menggunakan algoritma C-RNN mendapatkan error yang lebih rendah dibandingkan dengan CNN dan LSTM, yaitu mulai dari 510 – 530.
M.S. Islam, E. Hossain	Foreign exchange currency rate prediction using a GRU-LSTM hybrid network (2021)	GRU-LSTM, LSTM, GRU, SMA	Hasil menggunakan algoritma GRU-LSTM mendapatkan error yang lebih rendah
Gunho Jung, Sun-Yong Choi	Forecasting Foreign Exchange Volatility Using Deep Learning (2021)	LSTM, Autoencoder-LSTM	Untuk memprediksi valuta asing algoritma Autoencoder-LSTM mendapatkan error yang lebih rendah dibandingkan dengan LSTM
Aghistina Kartikadewi, Lina Audina Abdul Rosyid, Anggraeni Eka Putri	Prediction of Foreign Currency Exchange (IDR and USD) Using Multiple Linear Regression (2020)	Multiple Linear Regression	Dengan menggunakan model yang diajukan peneliti mendapatkan hasil kurang lebih 165,38% pada MSE, 24,04% pada MAPE, dan 25,7% pada margin error

Muhammad Yasir, Mehr Yahya Durrani, Sitara Afzal, Muazzam Maqsood, Farhan Aadil, Irfan Mehmood, Seungmin Rho	An Intelligent Event-Sentiment-Based Daily Foreign Exchange Rate Forecasting System (2019)	Linear Regression, SVR, Deep Learning	Hasilnya menunjukkan bahwa metode berbasis deep learning memiliki kinerja yang lebih baik daripada metode lainnya. Selain itu, hasil prediksi membaik ketika sentimen dipertimbangkan dalam model, oleh karena itu Hong Kong, Pakistan, dan Inggris dikatakan lebih terpapar pada peristiwa besar yang terjadi lintas batas.
Mei-Li Shen, Cheng-Feng Lee, Hsiou-Hsiang Liu, Po-Yin Chang, Cheng-Hong Yang	An Effective Hybrid Approach for Forecasting Currency Exchange Rates (2021)	FSPSOSVR, PSOSVR, SVR, ANN, SARIMA, ARIMA, EST, RW	Secara khusus, di bawah skema FSPSOSVR, MAPE-nya adalah 2,296%, mengungguli 3,477%, 4,628%, 3,603%, 4,657%, 4,333%, 6,018%, dan 4,089% dari skema milik PSOSVR, SVR, ANN, SARIMA, ARIMA, EST, dan RW
Manav Kaushik, A K Giri	Forecasting Foreign Exchange Rate: A Multivariate Comparative Analysis between Traditional Econometric, Contemporary Machine Learning & Deep Learning Techniques	VAR, SVM, LSTM	Hasilnya dengan jelas menggambarkan bahwa teknik kontemporer SVM dan RNN (Long Short-Term Memory) mengungguli metode tradisional Auto Regression yang banyak digunakan. Model RNN dengan Long Short-Term Memory (LSTM) memberikan akurasi maksimum (97,83%) diikuti oleh Model SVM (97,17%) dan Model VAR (96,31%).
Yaxin Qu, Xue Zhao	Application of LSTM Neural Network in Forecasting Foreign Exchange Price (2019)	LSTM, RNN	Hasil percobaan menunjukkan bahwa model jaringan saraf LSTM memiliki root mean square error (RMSE) dan mean absolute error (MAE) yang lebih kecil daripada model jaringan RNN, dan harga prediksi lebih akurat.
Ruofan Liao, Petchaluck Boonyakunakorn, Napat Harnpornchai,	Forecasting the Exchange Rate for USD to RMB using RNN and SVM (2020)	RNN, LM, SCG, BR, SVM, ARIMA	Hasilnya menunjukkan bahwa MSE terendah dimiliki oleh model RNN dibandingkan dengan LM, SCG, BR, SVM, ARIMA.

Songsak Sriboonchitta			
Kwok Tai Chui, Brij B. Gupta, Pandian Vasant	A Genetic Algorithm Optimized RNN-LSTM Model for Remaining Useful Life Prediction of Turbofan Engine (2021)	RNN, LSTM, NSGA-II optimized RNN-LSTM	Weight untuk RNN-LSTM yang dirancang oleh Non-Dominated Sorting Genetic Algorithm II (NSGA-II) dapat mencapai RMSE rata-rata 17,2. Ini meningkatkan RMSE sebesar 6,07–14,72% dibandingkan dengan model dasar RNN dan LSTM.
Azar Niknam, Hasan Khademi Zare, Hassan Hosseininassab, Ali Mostafaeipour	Developing an LSTM model to forecast the monthly water consumption according to the effects of the climatic factors in Yazd, Iran (2023)	UV-LSTM, MV-LSTM	Ditemukan bahwa kesalahan forecasting error MV-LSTM seringkali lebih kecil daripada model UV-LSTM. Ini berarti model MV-LSTM mengungguli UV-LSTM. Sedangkan, jika model memperhitungkan faktor iklim, akurasi peramalannya akan meningkat.
Burak Gülmez	Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm (2023)	LSTM-ARO, LSTM-GA, ANN, LSTM1D, LSTM2D, LSTM3D	Ketika LSTM-ARO dibandingkan dengan model artificial neural network (ANN), tiga model LSTM yang berbeda, dan LSTM yang dioptimalkan oleh Genetic Algorithm (GA). Hasilnya menunjukkan bahwa LSTM-ARO mengungguli model lain berdasarkan kriteria evaluasi MSE, MAE, MAPE, dan R2.

2.2. Landasan Teori

2.2.1. Valuta Asing

Nilai tukar mata uang nasional merupakan harga relatif terhadap mata uang nasional lainnya, dan seperti harga pada umumnya, nilai tukar dapat mengalami kenaikan atau penurunan. Ketika nilai tukar suatu mata uang, misalnya dolar, meningkat terhadap mata uang lain, seperti rupiah, hal ini menunjukkan bahwa satu unit mata uang tersebut dapat membeli lebih banyak mata uang lainnya. Dalam konteks ini, kita mengatakan bahwa mata uang tersebut menguat terhadap mata uang lainnya. Sebaliknya, ketika nilai tukar mata uang menurun terhadap mata uang lain, hal ini menunjukkan bahwa satu unit mata uang tersebut hanya dapat membeli jumlah mata uang lain yang lebih sedikit. Dalam hal ini, mata uang tersebut dianggap melemah terhadap mata uang lainnya [7].

2.2.2. Preprocessing

Data preprocessing atau *data preparation* adalah proses mengubah data mentah menjadi bentuk yang lebih sesuai untuk pemodelan. Tahap ini sering dianggap sebagai aspek yang paling krusial, memakan waktu, dan sering terlupakan dalam sebuah proyek pembelajaran mesin yang berfokus pada pemodelan prediktif. Meskipun prinsip dasar *data preparation* relatif sederhana, terdapat beragam teknik lanjutan yang masing-masing terdiri dari algoritma yang berbeda. Teknik-teknik ini secara khusus dirancang untuk mengatasi berbagai situasi, dan masing-masing memiliki sekumpulan *hyperparameter*, tips, dan trik mereka sendiri untuk mencapai hasil optimal [8].

2.2.2.1. Normalisasi

Normalisasi adalah proses mengubah rentang nilai aktual yang dapat diambil oleh fitur numerik menjadi rentang nilai standar yang biasanya dalam interval $[-1, 1]$ atau $[0, 1]$ [9]. Terdapat beberapa metode normalisasi yang umum digunakan, salah satunya adalah normalisasi *min-max* yang biasanya mengubah data dalam interval $[0, 1]$. Dimana persamaan normalisasi tersebut adalah sebagai berikut:

$$\bar{x}^{(j)} = \frac{x^{(j)} - \min^{(j)}}{\max^{(j)} - \min^{(j)}} \quad (2.1)$$

Keterangan:

$\bar{x}^{(j)}$ = Nilai hasil normalisasi

$x^{(j)}$ = Nilai fitur j

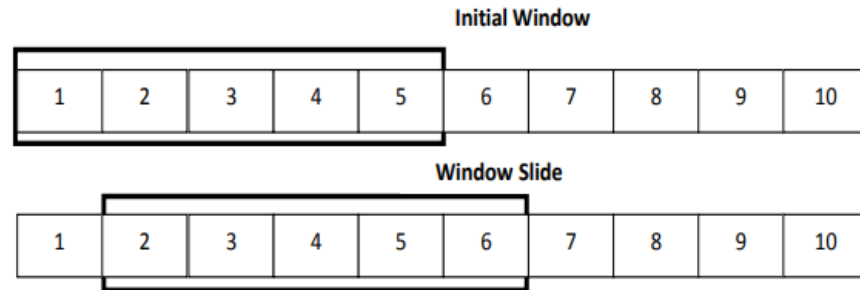
$\min^{(j)}$ = Nilai minimal dari fitur j

$\max^{(j)}$ = Nilai maksimal dari fitur j

2.2.2.2. Sliding Window

Sliding window merupakan salah satu metode yang dapat digunakan pada tahap *preprocessing* untuk merestrukturisasi data menurut kerangka waktu menjadi masalah klasifikasi [10]. Jumlah unit yang ditentukan dalam jendela disebut ukuran jendela. Setelah memilih segmen pertama, segmen berikutnya dipilih dari ujung segmen pertama.

Proses ini diulang sampai semua data deret waktu tersegmentasi. Proses *sliding window* ditunjukkan pada Gambar 2.1 dengan ukuran jendela 5.



Gambar 2.1 Proses Sliding Window (H.S. Hota dkk., 2017)

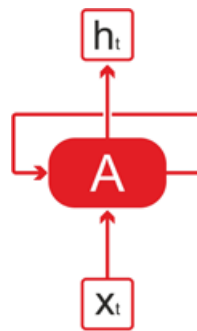
2.2.2.3. Split Data

Setelah memperoleh kumpulan data, langkah pertama yang dilakukan adalah melakukan pengacakan (*shuffle*) dan membagi data menjadi dua atau tiga bagian tergantung pada kebutuhan. Dalam era *Big Data* saat ini, umumnya data dibagi menjadi tiga bagian, yaitu: *training*, *validation*, dan *test*. Bagian *training* biasanya memiliki ukuran yang paling besar dan digunakan untuk melatih model. Sementara itu, bagian *validation* dan *test* memiliki ukuran yang relatif serupa dan jauh lebih kecil dibandingkan data *training*. Dimana *validation* digunakan untuk menyesuaikan *hyperparameter* model, dan *test* digunakan untuk mengevaluasi kinerja model pada data yang belum pernah dilihat sebelumnya [9].

2.2.3. Recurrent Neural Network (RNN)

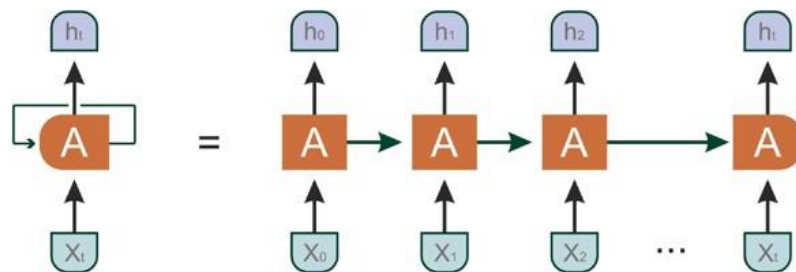
Recurrent Neural Networks (RNN) adalah salah satu jenis jaringan saraf yang dirancang untuk memproses *sequential data* dengan

memperkenalkan *loop* yang memungkinkan informasi bertahan di dalam jaringan. Tidak seperti jaringan saraf tradisional, yang hanya mempertimbangkan *input* saat ini, RNN dapat memanfaatkan informasi masa lalu untuk membuat prediksi atau mengklasifikasikan *input* saat ini [11].



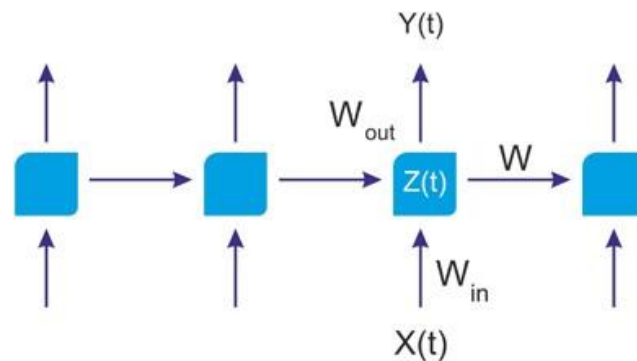
Gambar 2.2 RNN memiliki loop (G. Zaccane dkk., 2018)

Struktur dasar RNN terdiri dari modul berulang yang meneruskan pesan ke penggantinya. Saat dibuka, modul ini membuat struktur seperti rantai yang merepresentasikan aliran informasi sepanjang waktu. Setiap modul mengambil *input* pada langkah waktu tertentu dan menghasilkan *output*, sekaligus mempertahankan keadaan internal atau memori yang menangkap informasi tentang *input* sebelumnya.



Gambar 2.3 Representasi langkah dari RNN (G. Zaccane dkk., 2018)

Untuk mentransfer informasi antar langkah waktu, RNN menggunakan bobot transisi (W). Bobot ini memungkinkan jaringan untuk memperbarui status internalnya berdasarkan masukan saat ini dan status sebelumnya. Dengan demikian, RNN dapat menangkap dependensi dan pola dalam data berurutan.



Gambar 2.4 RNN menggunakan keadaan jaringan sebelumnya (G. Zaccane dkk., 2018)

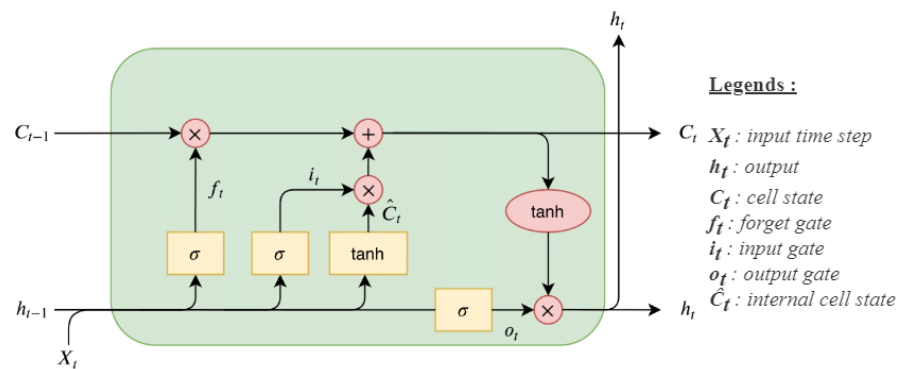
Namun, RNN klasik mengalami keterbatasan tertentu. Salah satu masalah utama adalah masalah gradien yang hilang, yang membuatnya sulit untuk menangkap ketergantungan jangka panjang. Selain itu, mereka kesulitan mempertahankan dan memanfaatkan informasi yang relevan dalam urutan yang panjang. Untuk mengatasi kelemahan ini, variasi RNN yang lebih baik yang disebut Long Short-Term Memory (LSTM) diperkenalkan.

2.2.4. Long Short Term Memory (LSTM)

Long Short Term Memory adalah jenis RNN khusus, yang mampu mempelajari dependensi jangka panjang. Layer tersebut diperkenalkan oleh Hochreiter & Schmidhuber pada tahun 1997 [11], yang bekerja sangat baik pada berbagai macam masalah dan sekarang

digunakan secara luas terutama dalam tugas yang melibatkan prediksi dan klasifikasi *sequential data*.

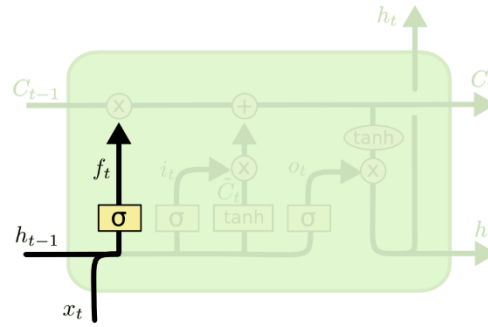
Jaringan LSTM terdiri dari sel atau blok yang saling berhubungan. Setiap blok berisi tiga jenis gerbang: *input*, *output*, dan *forget gate*. Gerbang ini mengontrol fungsi penulisan, pembacaan, dan pengaturan ulang pada sel memori.



Gambar 2.5 Arsitektur LSTM (Thorir, 2021)

2.2.4.1. Forget Gate

Forget gate menentukan berapa banyak data sebelumnya yang akan dilupakan dan berapa banyak data sebelumnya yang akan digunakan di langkah berikutnya. Hasil dari gerbang ini berada pada *range* 0-1. Nilai 0 melupakan data sebelumnya, 1 menggunakan data sebelumnya. *Forget gate* layer dapat dimodelkan seperti pada gambar 2.6. Dihitung dengan persamaan nomor 2.2.



Gambar 2.6 Forget Gate (Colah, 2015)

Persamaan Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.2)$$

Keterangan:

f_t = Forget gate

σ = Fungsi aktivasi sigmoid

W_f = Nilai weight forget gate

h_{t-1} = Nilai output sebelumnya

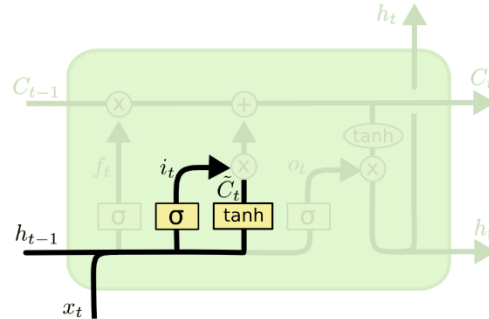
x_t = Nilai input saat ini

b_f = Nilai bias forget gate

2.2.4.2. Input Gate

Layer kedua adalah *input gate* yang terdiri dari *input gate* dan *tanh layer*. Data baru diperoleh pada lapisan ini. Bagian data *input* yang tidak diperlukan disaring dengan fungsi *sigmoid* dan kemudian data baru yang mungkin ditentukan dengan fungsi *tanh*. Perkalian hasil fungsi *sigmoid* dan hasil lapisan *tanh* ditambahkan ke keadaan sel untuk memperbaharui keadaan sel dan diperoleh keadaan sel yang baru. *Input*

gate dapat dimodelkan seperti pada gambar 2.7 dan 2.8. Dihitung dengan persamaan 2.3, 2.4, dan 2.5.



Gambar 2.7 Input Gate (Colah, 2015)

Persamaan Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.3)$$

Keterangan:

i_t = Input gate

σ = Fungsi aktivasi sigmoid

W_i = Nilai weight input gate

h_{t-1} = Nilai output sebelumnya

x_t = Nilai input saat ini

b_i = Nilai bias input gate

Persamaan Cell State baru

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.4)$$

Keterangan:

\hat{C}_t = Cell state baru

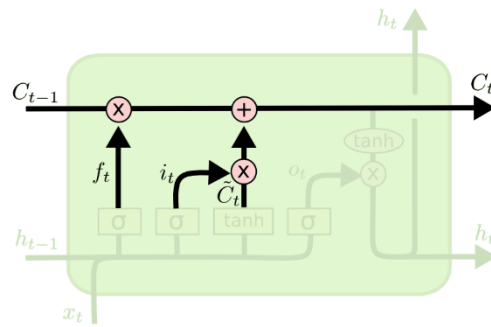
\tanh = Fungsi tanh

W_C = Nilai weight cell state

h_{t-1} = Nilai output sebelumnya

x_t = Nilai input saat ini

b_C = Nilai bias cell state



Gambar 2.8 Cell State (Colah, 2015)

Persamaan Memperbaharui Cell State

$$C_t = i_t \cdot \hat{C}_t + f_t \cdot C_{t-1} \quad (2.5)$$

Keterangan:

C_t = Cell state

i_t = Input gate

\hat{C}_t = Cell state baru

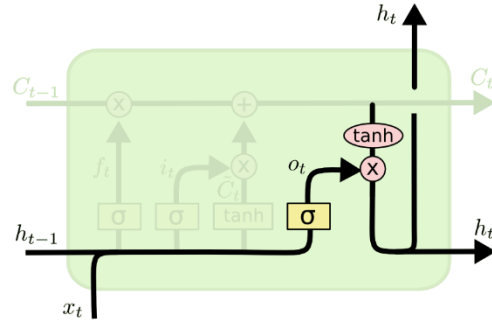
f_t = Forget gate

C_{t-1} = Cell state sebelumnya

2.2.4.3. Output Gate

Pada *output gate*, status sel difilter dengan menggunakan fungsi *tanh* dan data masukan difilter dengan fungsi *sigmoid*. Perkalian hasil fungsi *sigmoid* dengan hasil *tanh* layer menjadi data keluaran. *Output*

gate dapat dimodelkan seperti pada gambar 2.9. Dihitung dengan persamaan 2.6 dan 2.7.



Gambar 2.9 Output Gate (Colah, 2015)

Persamaan Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.6)$$

Keterangan:

o_t = Output gate

σ = Fungsi aktivasi sigmoid

W_o = Nilai weight output gate

h_{t-1} = Nilai output sebelumnya

x_t = Nilai input saat ini

b_o = Nilai bias output gate

Persamaan Nilai Output

$$h_t = o_t \times \tanh(C_t) \quad (2.7)$$

Keterangan:

h_t = Nilai output

o_t = Output gate

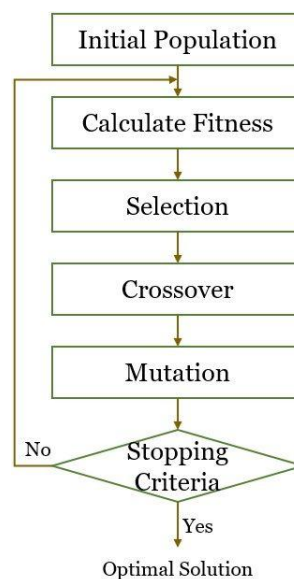
\tanh = Fungsi tanh

C_t = Cell state

2.2.5. Algoritma Genetik

Genetic Algorithm (GA) adalah pendekatan pencarian heuristik yang banyak digunakan untuk masalah optimisasi. Mereka fleksibel dan dapat diterapkan pada berbagai skenario pengoptimalan, menjadikannya menarik dalam aplikasi praktis. GA didasarkan pada konsep evolusi, menarik inspirasi dari keberhasilan dan keragaman spesies di alam [12].

Kemampuan beradaptasi spesies terhadap lingkungannya dan perkembangan struktur kompleks telah menjadi faktor kunci dalam kelangsungan hidup mereka. Prinsip-prinsip perkawinan dan menghasilkan keturunan merupakan dasar bagi keberhasilan evolusi. Dengan mengadaptasi prinsip-prinsip ini, GA bertujuan untuk memecahkan masalah pengoptimalan dengan meniru proses evolusi.



Gambar 2.10 Langkah Algoritma Genetik (Neha, 2022)

2.2.5.1. Fitness

Dalam GA, *fitness* merujuk pada ukuran kualitas suatu solusi. *Fitness function* digunakan untuk mengevaluasi setiap solusi kandidat berdasarkan kemampuannya dalam memecahkan masalah optimisasi. Desain *fitness function* merupakan bagian penting dari proses pemodelan pendekatan optimisasi, karena dapat membimbing pencarian. Sebagai contoh, dalam kasus masalah optimisasi yang terbatas, fungsi hukuman dapat digunakan untuk menurunkan *fitness* solusi yang tidak memenuhi syarat.

2.2.5.2. Seleksi

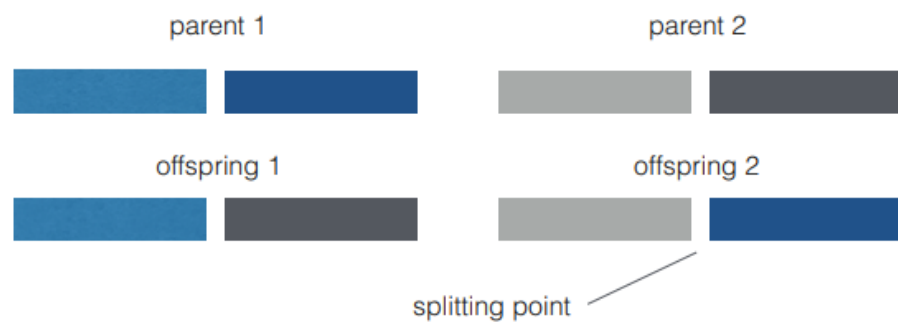
Seleksi adalah operator genetika dalam GA yang memilih solusi-solusi mana yang akan bertahan dan menjadi induk pada generasi baru. Proses seleksi didasarkan pada nilai kebugaran solusi-solusi dalam populasi, di mana solusi-solusi yang lebih baik memiliki peluang yang lebih tinggi untuk dipilih. Terdapat berbagai algoritma seleksi, salah satunya adalah *tournament selection*, di mana sekelompok solusi dipilih secara acak dan solusi-solusi terbaik dalam *subset* dipilih. Seleksi juga dapat digunakan menentukan induk – induk mana yang akan mengikuti proses *crossover*.



Gambar 2.11 Tournament Selection (A. Y. Ayoub dkk., 2020)

2.2.5.3. Crossover

Crossover adalah operator yang memungkinkan kombinasi materi genetik dari dua atau lebih solusi. Ini adalah operator genetik penting dalam GA, yang merupakan optimisasi heuristik yang diilhami secara biologis. Operator *crossover* dapat dirancang untuk berbagai jenis representasi solusi, seperti *bit strings*, *continuous vectors*, dan permutasi simbol. Salah satu contoh operator *crossover* untuk representasi *bit strings* adalah *crossover n-point*, yang membagi dua solusi pada posisi *n* dan secara bergantian menyusunnya menjadi solusi baru.

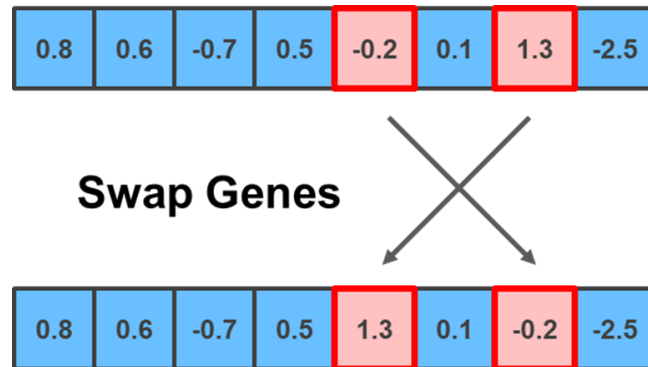


Gambar 2.12 Single-point Crossover (Kramer, 2017)

2.2.5.4. Mutasi

Mutasi adalah operator genetika penting lainnya dalam GA, yang mengubah sebuah solusi dengan memperkenalkan gangguan acak. Intensitas gangguan ini dikendalikan oleh tingkat mutasi. Operator mutasi harus memenuhi tiga persyaratan utama: keterjangkauan, ketidakberpihakan, dan skalabilitas. Berbagai operator mutasi dapat dirancang untuk berbagai jenis representasi solusi. Salah satunya adalah *swap mutation*, di mana setiap data akan ditukar dengan probabilitas

tertentu. Tingkat mutasi digunakan untuk mengatur intensitas dari *noise* yang ditambahkan.



Gambar 2.13 Swap Mutation (Silva, 2018)

2.2.6. Evaluasi Model

Evaluasi model adalah proses menggunakan matriks evaluasi yang berbeda untuk memahami kinerja model pembelajaran mesin, serta kekuatan dan kelemahannya. Evaluasi model penting untuk menilai kemandirian model selama fase penelitian awal dan juga berperan dalam pemantauan model. Dalam pembuatan *regression* model, matriks evaluasi yang digunakan adalah matriks yang dapat menghitung *error*, antara lain MAE, MSE, RMSE, dan lain – lain.

2.2.6.1. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) adalah rata-rata dari selisih absolut antara nilai yang diamati dan nilai yang diprediksi. MAE juga dikenal sebagai Mean Absolute Deviation. Perbedaan antara MAE dan MSE adalah bahwa MAE mengambil selisih absolut antara nilai yang diprediksi dan nilai aktual, sedangkan MSE mengambil selisih kuadrat [13]. Persamaan untuk MAE adalah sebagai berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (2.8)$$

Keterangan:

MAE = Mean Absolute Error

n = Jumlah data

\hat{y}_i = Nilai prediksi

y_i = Nilai aktual

2.2.6.2. Mean Squared Error (MSE)

Mean Squared Error (MSE), juga dikenal sebagai Mean Squared Deviation, adalah perbedaan kuadrat antara nilai aktual dan nilai yang diprediksi. MSE mengindikasikan sejauh mana garis terbaik cocok dengan sekumpulan titik data. MSE selalu memiliki nilai positif. Penggunaan kuadrat dilakukan untuk menghilangkan tanda negatif. Semakin mendekati nilai MSE ke 0, prediksi menjadi semakin akurat [13]. Persamaan untuk MSE adalah sebagai berikut:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (2.9)$$

Keterangan:

MSE = Mean Square Error

n = Jumlah data

\hat{y}_i = Nilai prediksi

y_i = Nilai aktual

2.2.6.3. Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) adalah akar kuadrat dari rata-rata kuadrat dari semua *error*. RMSE juga dikenal sebagai Root Mean Squared Deviation. Dengan kata lain, RMSE adalah standar deviasi dari *error*. RMSE juga mengindikasikan sejauh mana garis terbaik cocok dengan sekumpulan titik data [13]. Persamaan untuk RMSE adalah sebagai berikut:

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}} = \sqrt{MSE} \quad (2.10)$$

Keterangan:

RMSE = Root Mean Squared Error

n = Jumlah data

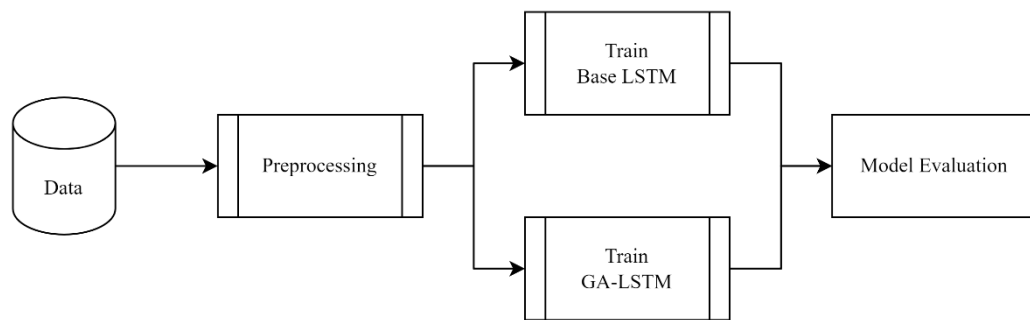
\hat{y}_i = Nilai prediksi

y_i = Nilai aktual

MSE = Mean Squared Error

BAB III

METODE PENELITIAN



Gambar 3.1 Langkah Penelitian

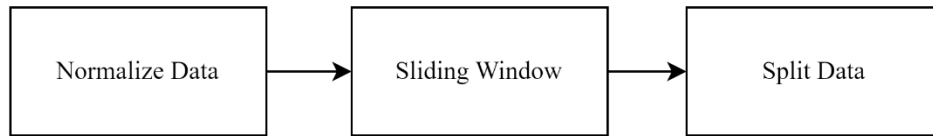
3.1. Deskripsi Data

Data yang akan peneliti gunakan untuk penelitian ini adalah data harga beli dari sebuah valuta asing setiap harinya. Data yang peneliti ambil memiliki rentang waktu kurang lebih 3 tahun sebelumnya, mulai dari 1 Januari 2020 – 1 Juni 2023. Valuta asing yang peneliti gunakan adalah USD/IDR, EUR/IDR, dan JPY/IDR. Dimana tiap – tiap data tersebut hanya memiliki 2 atribut, yaitu tanggal dan harga beli. Data – data tersebut diambil dari platform Google Finance dengan menggunakan *function* yang sudah disediakan oleh Google Spreadsheet.

Tabel 3.1 Contoh Data Mentah

Date	Close
01/01/2020 23:58:00	13689.23
02/01/2020 23:58:00	13884.79
03/01/2020 23:58:00	13935.46

3.2. Preprocessing



Gambar 3.2 Langkah Preprocessing

3.2.1. Normalisasi Data

Langkah pertama *preprocessing* data valuta asing adalah normalisasi. Normalisasi yang akan digunakan adalah normalisasi *min-max*. Dimana tiap – tiap fitur valuta asing tersebut di normalisasi menggunakan persamaan 2.1 dan hasilnya seperti yang terdapat pada tabel 3.2.

Tabel 3.2 Contoh Data Normalisasi

Sebelum Normalisasi	Sesudah Normalisasi
13689.23	0
13884.79	0.794217
13935.46	1

3.2.2. Sliding Window

Langkah selanjutnya adalah menyegmentasi data menggunakan *sliding window*. Dimana data – data tersebut akan disegmentasikan berdasarkan ukuran jendela. Peneliti memutuskan untuk menggunakan nilai 99 sebagai ukuran jendela.

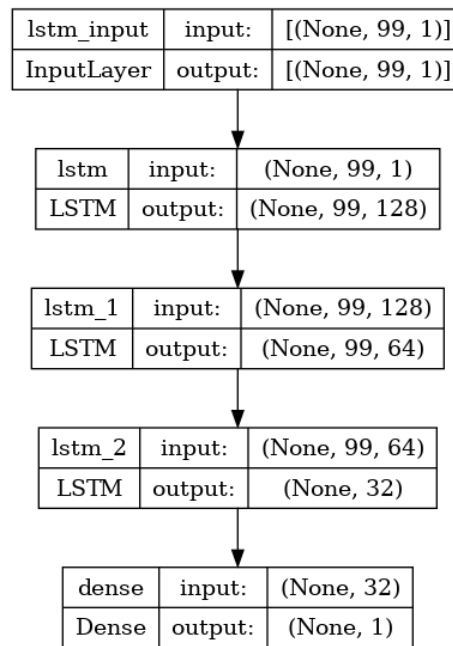
3.2.3. Split Data

Langkah selanjutnya adalah membagi data menjadi 3 bagian, yaitu, *training*, *validation*, dan *test*. Data tersebut dibagi menggunakan komposisi 9:1, 90% untuk *training*, 10% untuk *test*. Sedangkan untuk

bagian *validation* mereka akan otomatis terbuat jika memasukkan parameter saat mentraining model. Ukurannya kurang lebih adalah 10% dari data *training*.

3.3. Implementasi Model

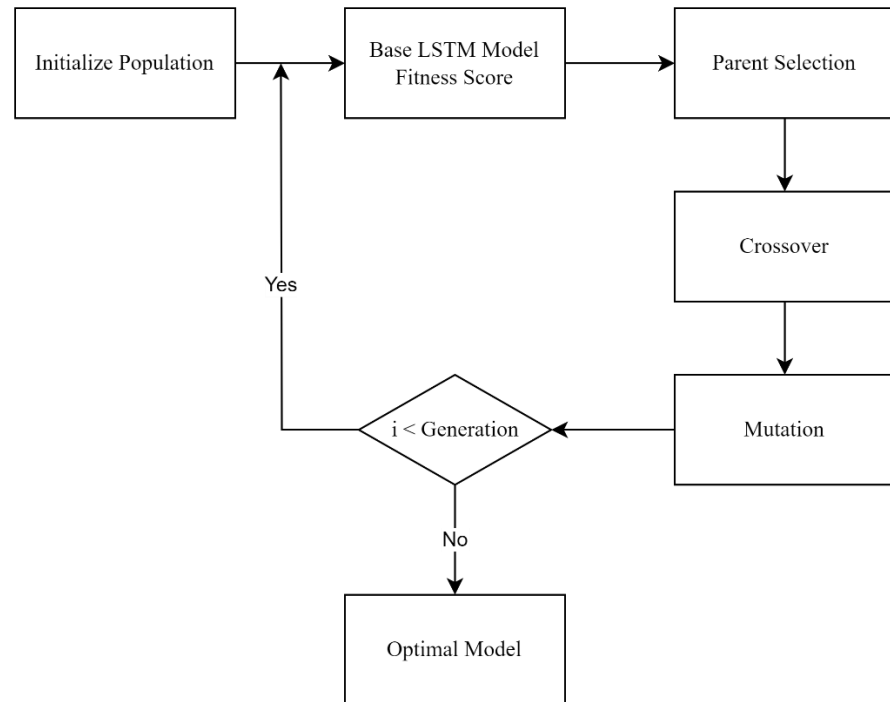
3.3.1. Base LSTM



Gambar 3.3 Arsitektur LSTM

Gambar 3.3 menunjukkan standar LSTM model arsitektur. Arsitektur model yang peneliti gunakan terdiri atas *Input layer* dengan ukuran sebesar *sliding window*. Dilanjutkan dengan 3 LSTM *layer* yang memiliki ukuran 128, 64, dan 32. Dimana nanti 2 layer LSTM terakhir, 64 dan 32, akan dibuka secara bergantian saat percobaan. Terakhir, hasil dikeluarkan oleh *Dense output layer* dengan ukuran 1.

3.3.2. GA-LSTM



Gambar 3.4 Langkah GA-LSTM

Gambar 3.4 menunjukkan langkah bagaimana GA mengoptimalkan parameter yang ada di LSTM. Arsitektur model yang digunakan juga sama seperti gambar 3.3. Dimana nanti 2 layer LSTM terakhir, 64 dan 32, akan dibuka secara bergantian saat percobaan.

a. Inisiasi Populasi

Langkah pertama adalah menginisiasi populasi secara acak. Dimana populasi tersebut akan berbentuk *list* 2 dimensi, dengan panjang baris sepanjang apa yang dimasukkan oleh peneliti dan 3 kolom yang menentukan parameter dari *layer* LSTM.

b. Menghitung Fitness

Langkah kedua adalah menghitung nilai *fitness*. Nilai *fitness* didapatkan setiap kali model dilatih dengan kromosom atau setiap data yang ada pada populasi dan dievaluasi dengan data *test*.

c. Seleksi

Langkah ketiga adalah seleksi kromosom yang nanti akan di jadikan induk pada saat *crossover*. Tipe seleksi yang peneliti gunakan adalah *tournament selection*.

d. Crossover

Langkah keempat adalah melakukan *crossover* terhadap kromosom yang telah seleksi. Tipe *crossover* yang peneliti gunakan adalah *single point crossover*.

e. Mutasi

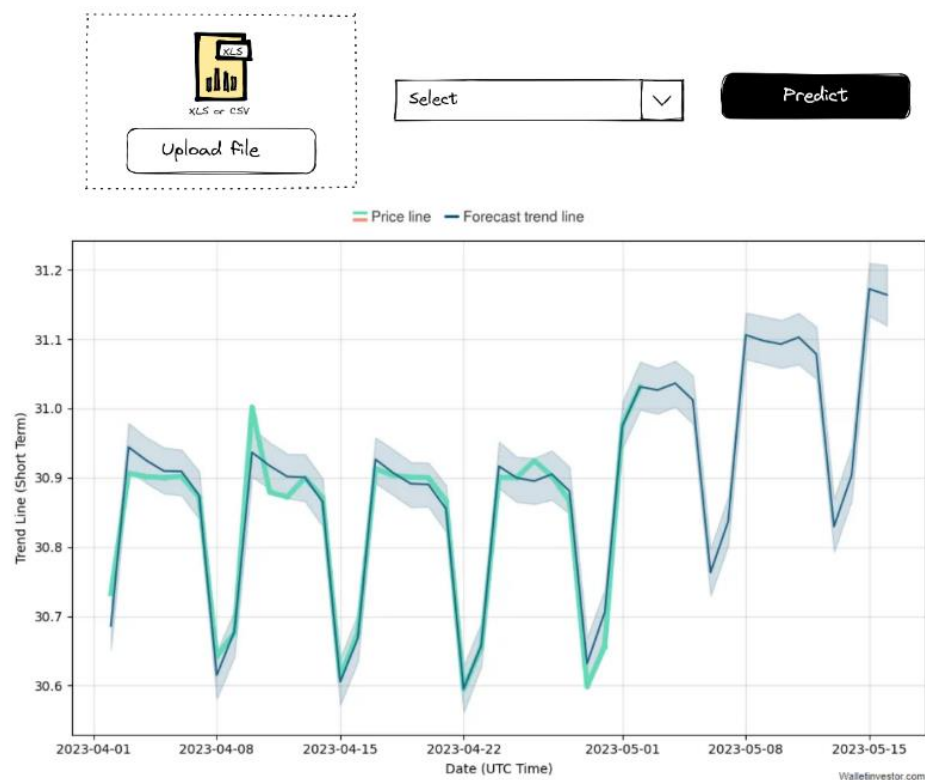
Langkah kelima adalah melakukan mutasi terhadap hasil *crossover*. Tipe mutasi yang peneliti gunakan adalah *swap mutation*. Dimana mutasi tersebut akan dijalankan dengan probabilitas 1%.

Terakhir, langkah b sampai dengan e akan diulang terus menerus sampai dengan generasi yang dimasukkan oleh peneliti. Setelah iterasi selesai, GA akan mengembalikan nilai yang paling optimal sebagai parameter model LSTM.

3.4. Evaluasi Model

Dalam penelitian ini kedua model tersebut akan dievaluasi dengan data yang belum pernah dilihat. Hasil prediksi sebuah model dan nilai aktual dapat mendapatkan nilai bagaimana performa model tersebut dalam memprediksi data. Nilai performa model tersebut dapat dihitung menggunakan persamaan 2.8, 2.9, dan 2.10.

3.5. Desain User Interface



Gambar 3.5 Rancangan Desain GUI

Rancangan desain GUI yang peneliti ajukan kurang lebih seperti di atas. Untuk langkah penggunaannya kurang lebih seperti ini:

1. Persiapkan data harga valuta asing USD/IDR, EUR/IDR, atau JPY/IDR

2. Pastikan data berformat csv dan hanya memiliki 2 kolom , yaitu, tanggal dan harga
3. User mengupload data berformat CSV
4. User memilih waktu data itu ingin diprediksi ke masa depan, mulai dari 1 hari, 1 minggu, atau 1 bulan
5. Langkah terakhir, menekan tombol *predict*
6. Tunggu program selesai berjalan dan hasilnya akan ditampilkan dalam bentuk grafik

3.6. Kebutuhan Hardware dan Software

a. Spesifikasi Hardware

1. Intel(R) Core(TM) i7-11800H @ 2.30GHz
2. NVIDIA GeForce RTX 3050Ti
3. RAM 16 GB
4. SSD

b. Spesifikasi Software

1. Windows 11
2. Visual Studio Code (Basic LSTM)
3. Kaggle (GA-LSTM)
4. Python 3.10
5. Library Python:
 - TensorFlow dan Keras
 - Sklearn
 - Pandas
 - Numpy
 - Matplotlib

3.7. Rancangan Skenario Pengujian

Tabel 3.3 Skenario Pengujian

Skenario		LSTM	GA-LSTM
1	<i>LSTM Layer:</i> 1	Epoch: 20	Populasi: 30 Generasi: 20
2		Epoch: 50	Populasi: 30 Generasi: 50
3	<i>LSTM Layer:</i> 2	Epoch: 20	Populasi: 30 Generasi: 20
4		Epoch: 50	Populasi: 30 Generasi: 50
5	<i>LSTM Layer:</i> 3	Epoch: 20	Populasi: 30 Generasi: 20
6		Epoch: 50	Populasi: 30 Generasi: 50

JADWAL KEGIATAN

No.	Bulan	Kegiatan
1	Juni	Pengumpulan Data
2	Juli	Skenario 1 – 2
3	Agustus	Skenario 3 – 4
4	September	Skenario 5 – 6
5	Oktober	Evaluasi
6	November	Pengerjaan Bab IV
7	Desember	Pengerjaan Bab V

DAFTAR PUSTAKA

- [1] M. S. Islam dan E. Hossain, "Foreign exchange currency rate prediction using a GRU-LSTM hybrid network," *ELSEVIER*, no. 3, 2021.
- [2] A. Kartikadewi, L. A. A. Rosyid dan A. E. Putri, "Prediction of Foreign Currency Exchange (IDR and USD) Using Multiple Linear Regression," *International Journal of Engineering and Techniques*, vol. VI, no. 2, 2020.
- [3] N. Lina, L. Yujie, W. Xiao, Z. Jinqun, Y. Jiguo dan Q. Chengming, "Forecasting of Forex Time Series Data Based on Deep Learning," *ELSEVIER*, no. 147, pp. 647-652, 2019.
- [4] Z. Hu, Y. Zhao dan M. Khushi, "A Survey of Forex and Stock Price Prediction Using Deep Learning," *Appl. Syst. Innov.*, vol. IV, no. 9, 2021.
- [5] M. Yasir, M. Y. Durrani, S. Afzal, M. Maqsood, F. Aadil, I. Mehmood dan S. Rho, "An Intelligent Event-Sentiment-Based Daily Foreign Exchange Rate Forecasting System," *Applied Science*, vol. IX, no. 15, p. 2980, 2019.
- [6] Q. Yaxin dan Z. Xue, "Application of LSTM Neural Network in Forecasting Foreign Exchange Price," *Journal of Physics: Conference Series*, vol. 1237, no. 4, 2019.
- [7] J. A. Frieden, D. A. Lake dan K. A. Schultz, *World Politics: Interests, Interactions, Institutions 4th Edition*, New York: W.W. Norton & Company, 2019.
- [8] J. Brownlee, *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*, 2020.
- [9] A. Burkov, *The Hundred-Page Machine Learning Book*, 2019.
- [10] N. M. Norwawi, "Sliding window time series forecasting with multilayer perceptron and multiregression of COVID-19 outbreak in Malaysia," *ELSEVIER*, pp. 547-564, 2021.
- [11] G. Zaccane dan M. R. Karim, *Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python*, 2nd Edition, Birmingham: Packt Publishing, 2018.
- [12] O. Kramer, *Genetic Algorithm Essentials*, Oldenburg: Springer Nature, 2017.

- [13] A. V. Tatachar, "Comparative Assessment of Regression Models Based On Model," *International Research Journal of Engineering and Technology (IRJET)*, vol. 08, no. 09, 2021.