

BAB 3

METODOLOGI PENELITIAN

3.1 Spesifikasi Sistem

Dalam proses penelitian ini digunakan spesifikasi sistem sebagai berikut:

1. *Software*

- *Google Chrome*
- *Jupyter Notebook*
- *Operating System* : Windows 11

2. *Hardware*

- *Processor* : Intel I5-10300H
- RAM: 16 GB
- GPU: GTX 1650

3.2 Alur Penelitian

1. Studi literatur

Pembelajaran terhadap teori yang digunakan dalam pengembangan sistem prediksi saham menggunakan LSTM dengan tambahan fitur *weighted exponential moving average*.

2. Pengumpulan data

Pengumpulan data dari *Yahoo Finance* dengan *library Python YFinance* untuk data historis saham yang digunakan pada penelitian ini.

3. Praproses data

Sebelum dataset digunakan pada model LSTM perlu dilakukan praproses data atau mengolah data terlebih dahulu agar LSTM mampu memprediksi data dengan efektif dan akurat. Tahapan praproses data dimulai dengan pemilihan fitur data, pembagian dataset menjadi *training*, *validation*, dan *testing*, normalisasi data, dan *reshape* data menjadi bentuk tiga dimensi.

4. *Grid Search* untuk optimasi *hyperparameter*

Proses *tuning* untuk mencari *hyperparameter* terbaik dengan mencoba seluruh kombinasi dari *hyperparameter* yang telah disediakan dan dilakukan pengecekan apakah mendapatkan nilai MAPE untuk prediksi data validasi terbaik.

5. Pembuatan model LSTM dengan *hyperparameter* yang sudah optimal

Membuat model LSTM yang digunakan pada penelitian ini dengan *hyperparameter* terbaik untuk setiap emiten yang diprediksi.

6. Evaluasi dengan RMSE dan MAPE

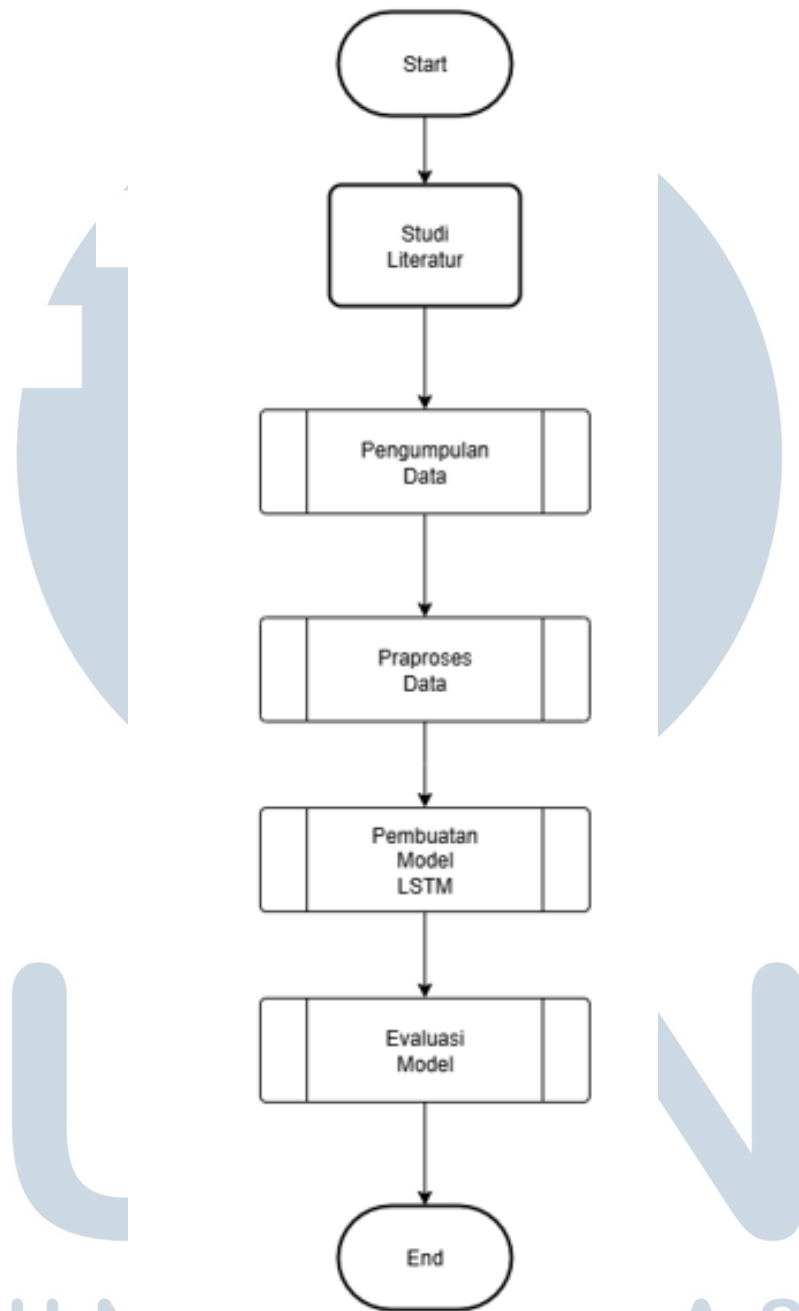
Pengujian kinerja model dengan memprediksi data yang belum pernah dilihat sebelumnya dan dinilai dengan metode *Root Mean Squared Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE).

3.3 Studi Literatur

Pada tahap ini dilakukan pembelajaran terhadap teori yang akan digunakan dalam pengembangan sistem prediksi harga saham. Teori yang akan dipelajari adalah *Weighted Exponential Moving Average*, algoritma *Recurrent Neural Network*, algoritma *Gated Recurrent Unit*, algoritma *Long Short-Term Memory* (LSTM), RMSE dan MAPE untuk mengevaluasi model, *min-max normalization* untuk normalisasi, *Grid Search* sebagai metode untuk mencari *hyperparameter* model terbaik, serta *Holdout Validation*. Sumber teori pembelajaran didapat dari jurnal, buku, dan penelitian yang telah dilakukan sebelumnya.

3.4 Flowchart Metodologi Penelitian

Alur penelitian dapat dilihat pada Gambar 3.1. Metodologi penelitian dimulai dengan pencarian teori yang relevan dengan penelitian pada tahap studi literatur, selanjutnya tahap pengumpulan data yang diambil dari dataset yfinance, kemudian dilakukan tahap preprocessing untuk mempersiapkan data sebelum diproses oleh model LSTM dan dilanjutkan oleh pembuatan model LSTM serta evaluasi model LSTM penelitian ini.

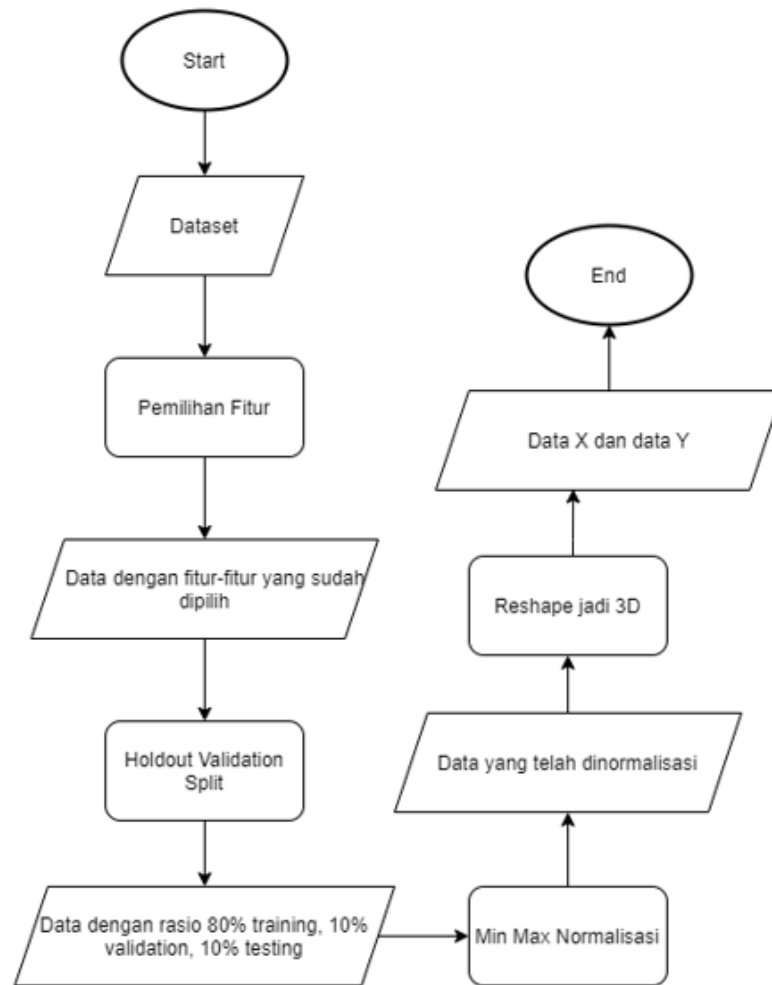


Gambar 3.1. *Flowchart* metodologi penelitian

3.5 Flowchart Pre-Processing

Alur tahap *pre-processing* dapat dilihat pada Gambar 3.2. Tahap ini dimulai dari pemilihan fitur-fitur dataset yang digunakan untuk model penelitian, kemudian dari data dengan fitur-fitur yang sudah dipilih dibagi dengan metode *holdout*

validation split dengan rasio 80 persen *training*, 10 persen *validation* dan 10 persen *testing*. Setelah itu, data dinormalisasi menggunakan *min max normalization* berdasarkan dengan data *training*. Terakhir data diubah menjadi bentuk tiga dimensi dan dijadikan sebagai X untuk input dan Y untuk target atau *output* LSTM.

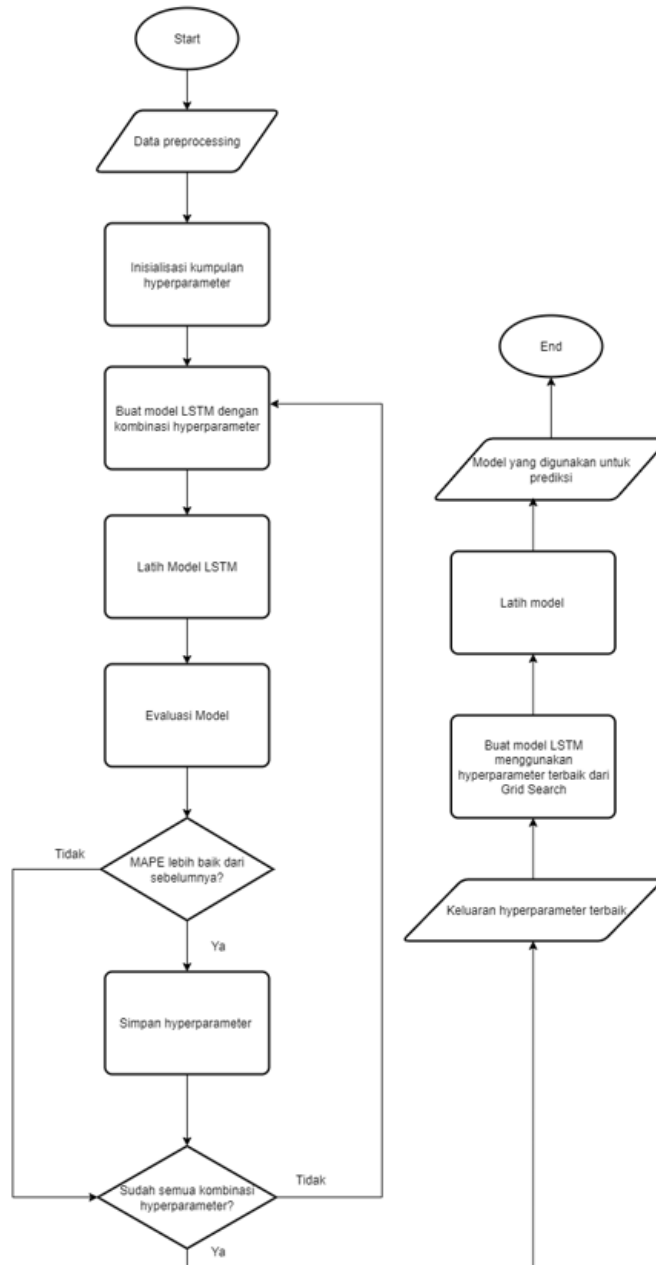


Gambar 3.2. Flowchart Pre-Processing

3.6 Flowchart Main-Processing

Alur tahap *main-processing* dapat dilihat pada Gambar 3.3. Tahap ini dimulai dengan inialisasi kumpulan hyperparameter yang akan dicari kombinasi terbaiknya dengan metode *Grid Search*. Saat *Grid Search*, seluruh kombinasi dari *hyperparameter* digunakan untuk model dan dievaluasi nilai MAPE nya berdasarkan hasil prediksi menggunakan data validasi. Saat mendapat MAPE

terbaik maka kombinasi *hyperparameter* disimpan. Setelah semua tahap itu, maka model LSTM dengan *hyperparameter* terbaik dibuat dan dilatih. Model ini kemudian dipakai untuk prediksi data *testing*.

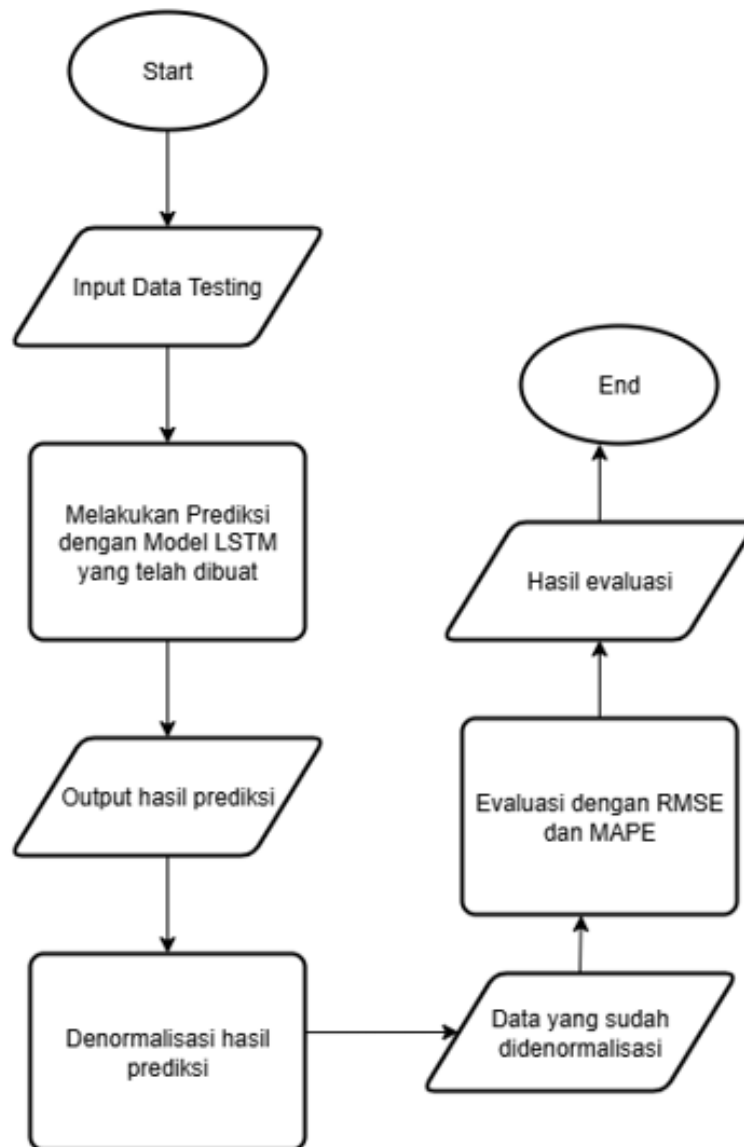


Gambar 3.3. Flowchart Main-Processing

3.7 Flowchart Post-Processing

Alur tahap *post-processing* dapat dilihat pada Gambar 3.4. Tahap ini dilakukan setelah model telah dibuat dan dilatih. Dimulai dengan memasukkan data testing atau data yang belum pernah dilihat model sebelumnya dalam bentuk yang sudah dinormalisasi, kemudian model mengeluarkan hasil prediksi masih dalam format normalisasi. Hasil prediksi sebelum dievaluasi harus didenormalisasi agar format data sesuai dengan data aslinya. Selanjutnya, dilakukan evaluasi dengan metode RMSE atau *Root Mean Squared Error* dan MAPE atau *Mean Absolute Percentage Error* dan hasil evaluasi ditampilkan dalam bentuk tabel.





Gambar 3.4. Flowchart Post-Processing

3.8 Pengumpulan Data

Dataset yang digunakan untuk pembuatan sistem prediksi harga saham LSTM didapat dari Yahoo Finance dengan *library* Python *yfinance*. Data yang diambil adalah data historis saham dari 24 Februari 2015 sampai 21 Februari 2025. Emiten yang diprediksi berada pada indeks LQ45 dan bergerak di sektor energi seperti ADRO, PGAS, dan PTBA.

3.9 Praproses Data

Sebelum dataset digunakan pada model LSTM perlu dilakukan praproses data atau mengolah data terlebih dahulu agar LSTM mampu memprediksi data dengan efektif dan akurat. Tahapan praproses data dimulai dengan pemilihan fitur yang digunakan model LSTM, pembagian dataset menjadi *training*, *validation* dan *testing*, normalisasi untuk menyesuaikan format data agar data dengan nilai yang tinggi tidak mendominasi proses pelatihan LSTM, dan terakhir bentuk data diubah atau *reshape* untuk menyesuaikan format yang dapat digunakan pada LSTM.

3.9.1 Pemilihan Fitur

Dari data yang diambil dataset yfinance terdapat beberapa fitur yang digunakan untuk prediksi saham menggunakan LSTM penelitian ini. Fitur-fitur yang digunakan antara lain adalah harga penutupan, harga pembukaan, harga tertinggi harian, harga terendah harian, WEMA 5 hari dan WEMA 50 hari dihitung menggunakan Persamaan 2.2. WEMA atau *Weighted Exponential Moving Average* adalah *moving average* yang digunakan untuk LSTM lebih baik belajar dari tren saham sektor energi. Penggunaan WEMA diambil dari penelitian [8]. Ketika *moving average* jangka pendek berada di atas *moving average* jangka panjang maka tren saham cenderung naik, sebaliknya ketika *moving average* jangka pendek berada di bawah *moving average* jangka panjang maka tren saham cenderung menurun [10]. Penggunaan banyak fitur menghasilkan dapat menghasilkan performa yang lebih baik berdasarkan dari penelitian [30]. Grafik harga penutupan saham dan *weighted exponential moving averages* dapat dilihat pada Gambar 3.5, 3.6, dan 3.7.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.5. Harga penutupan dan *Weighted Exponential Moving Average* emiten ADRO



Gambar 3.6. Harga penutupan dan *Weighted Exponential Moving Average* emiten PGAS



Gambar 3.7. Harga penutupan dan *Weighted Exponential Moving Average* emiten PTBA

3.9.2 Pembagian Dataset

Total dataset sebesar 2472 hari perdagangan saham untuk masing-masing emiten yang kemudian dibagi menjadi *training*, *validation*, dan *testing* dengan rasio pembagian 80%, 10%, 10%. Pembagian dengan rasio 80-10-10 umum digunakan di *machine learning* [28]. Pembagian ini menyediakan cukup data *training* untuk pelatihan model, data validasi digunakan untuk mencari *hyperparameter* terbaik pada model dan data *testing* dipakai untuk menguji performa model terhadap data yang belum pernah dilihat sebelumnya. Pembagian ini penting untuk mencegah terjadinya *overfitting* dan *underfitting*. *Overfitting* terjadi ketika model terlalu banyak berlatih dengan data *training* sehingga akurasi pada data *testing* berkurang. Sebaliknya, *underfitting* terjadi karena model kurang lama dalam proses pelatihan menggunakan data *training* sehingga tidak sempat mempelajari pola-pola penting dalam data dan tingkat akurasi berkurang.

Data *training* yang terdiri dari 80 persen dataset atau 1977 baris data untuk masing-masing emiten dapat dilihat pada Gambar 3.8, 3.9, dan 3.10. Rentang waktu untuk data *training* dimulai dari tanggal 24 Februari 2015 sampai dengan 20 Januari 2023. Data ini dipakai saat proses pelatihan model untuk memperbarui *weight* dan *bias*.

Training Data ADRO.JK						
	Closing	WEMA 5	WEMA 50	Open	Low	High
Date						
2015-02-24 00:00:00+07:00	965.0	980.11	964.45	980.0	955.0	980.0
2015-02-25 00:00:00+07:00	960.0	980.11	964.45	960.0	950.0	975.0
2015-02-26 00:00:00+07:00	945.0	980.11	964.45	965.0	940.0	970.0
2015-02-27 00:00:00+07:00	960.0	980.11	964.45	945.0	940.0	965.0
2015-03-02 00:00:00+07:00	1020.0	980.11	964.45	970.0	965.0	1025.0
...	...	...	...	...	...	...
2023-01-16 00:00:00+07:00	3100.0	3103.11	3685.04	3120.0	3090.0	3130.0
2023-01-17 00:00:00+07:00	3140.0	3138.67	3679.35	3060.0	3050.0	3150.0
2023-01-18 00:00:00+07:00	3170.0	3146.44	3675.44	3160.0	3150.0	3230.0
2023-01-19 00:00:00+07:00	3230.0	3154.89	3676.87	3140.0	3140.0	3230.0
2023-01-20 00:00:00+07:00	3240.0	3172.89	3682.19	3250.0	3210.0	3260.0
1977 rows × 6 columns						

Gambar 3.8. Data Pelatihan ADRO

Training Data PGAS.JK						
	Closing	WEMA 5	WEMA 50	Open	Low	High
Date						
2015-02-24 00:00:00+07:00	5475.0	5360.00	5008.96	5425.0	5400.0	5500.0
2015-02-25 00:00:00+07:00	5375.0	5360.00	5008.96	5500.0	5325.0	5500.0
2015-02-26 00:00:00+07:00	5300.0	5360.00	5008.96	5400.0	5275.0	5400.0
2015-02-27 00:00:00+07:00	5200.0	5360.00	5008.96	5275.0	5200.0	5300.0
2015-03-02 00:00:00+07:00	5300.0	5360.00	5008.96	5200.0	5200.0	5300.0
...	...	...	...	...	...	...
2023-01-16 00:00:00+07:00	1540.0	1541.56	1811.95	1540.0	1525.0	1555.0
2023-01-17 00:00:00+07:00	1535.0	1541.22	1808.90	1550.0	1535.0	1565.0
2023-01-18 00:00:00+07:00	1570.0	1544.89	1806.99	1545.0	1540.0	1585.0
2023-01-19 00:00:00+07:00	1585.0	1552.56	1805.03	1560.0	1555.0	1595.0
2023-01-20 00:00:00+07:00	1620.0	1570.67	1804.84	1595.0	1585.0	1630.0
1977 rows x 6 columns						

Gambar 3.9. Data Pelatihan PGAS

Training Data PTBA.JK						
	Closing	WEMA 5	WEMA 50	Open	Low	High
Date						
2015-02-24 00:00:00+07:00	2080.0	2133.44	2106.01	2090.0	2070.0	2145.0
2015-02-25 00:00:00+07:00	2085.0	2133.44	2106.01	2100.0	2070.0	2110.0
2015-02-26 00:00:00+07:00	2085.0	2133.44	2106.01	2100.0	2080.0	2110.0
2015-02-27 00:00:00+07:00	2135.0	2133.44	2106.01	2085.0	2080.0	2135.0
2015-03-02 00:00:00+07:00	2200.0	2133.44	2106.01	2150.0	2150.0	2210.0
...	...	...	...	...	...	...
2023-01-16 00:00:00+07:00	3320.0	3330.22	3650.53	3320.0	3300.0	3350.0
2023-01-17 00:00:00+07:00	3350.0	3355.33	3644.23	3310.0	3300.0	3390.0
2023-01-18 00:00:00+07:00	3410.0	3372.22	3640.27	3360.0	3360.0	3460.0
2023-01-19 00:00:00+07:00	3490.0	3387.33	3641.81	3410.0	3410.0	3490.0
2023-01-20 00:00:00+07:00	3570.0	3425.11	3650.53	3500.0	3480.0	3570.0
1977 rows x 6 columns						

Gambar 3.10. Data Pelatihan PTBA

Selanjutnya, data *validation* terdiri 10 persen dataset atau 247 baris dimulai dari rentang waktu 24 Januari 2023 sampai dengan satu Februari 2024. Data ini

dipakai untuk mencari *hyperparameter* terbaik pada saat *tuning hyperparameter* dengan *Grid Search*. Data *validation* dapat dilihat pada Gambar 3.11, 3.12, dan 3.13.

Validation Data ADRO.JK						
	Closing	WEMA 5	WEMA 50	Open	Low	High
Date						
2023-01-24 00:00:00+07:00	3200.0	3200.0	3684.97	3240.0	3170.0	3250.0
2023-01-25 00:00:00+07:00	3130.0	3130.0	3685.98	3190.0	3120.0	3190.0
2023-01-26 00:00:00+07:00	3010.0	3010.0	3684.37	3050.0	2990.0	3070.0
2023-01-27 00:00:00+07:00	3060.0	3060.0	3686.73	2980.0	2980.0	3080.0
2023-01-30 00:00:00+07:00	3010.0	3010.0	3684.03	3060.0	2980.0	3100.0
...	...	...	...	...	...	...
2024-01-26 00:00:00+07:00	2380.0	2380.0	2528.20	2360.0	2330.0	2390.0
2024-01-29 00:00:00+07:00	2400.0	2400.0	2529.95	2380.0	2370.0	2410.0
2024-01-30 00:00:00+07:00	2390.0	2390.0	2529.31	2390.0	2360.0	2400.0
2024-01-31 00:00:00+07:00	2400.0	2400.0	2528.59	2410.0	2390.0	2450.0
2024-02-01 00:00:00+07:00	2420.0	2420.0	2527.79	2420.0	2390.0	2450.0

247 rows × 6 columns

Gambar 3.11. Data Validasi ADRO

Validation Data PGAS.JK						
	Closing	WEMA 5	WEMA 50	Open	Low	High
Date						
2023-01-24 00:00:00+07:00	1595.0	1595.0	1801.38	1635.0	1585.0	1650.0
2023-01-25 00:00:00+07:00	1575.0	1575.0	1798.34	1600.0	1575.0	1605.0
2023-01-26 00:00:00+07:00	1560.0	1560.0	1795.69	1575.0	1555.0	1600.0
2023-01-27 00:00:00+07:00	1565.0	1565.0	1793.46	1565.0	1560.0	1590.0
2023-01-30 00:00:00+07:00	1565.0	1565.0	1791.05	1570.0	1560.0	1590.0
...	...	...	...	...	...	...
2024-01-26 00:00:00+07:00	1130.0	1130.0	1122.95	1145.0	1125.0	1145.0
2024-01-29 00:00:00+07:00	1130.0	1130.0	1122.74	1135.0	1120.0	1140.0
2024-01-30 00:00:00+07:00	1115.0	1115.0	1121.93	1130.0	1115.0	1130.0
2024-01-31 00:00:00+07:00	1165.0	1165.0	1123.48	1120.0	1120.0	1175.0
2024-02-01 00:00:00+07:00	1160.0	1160.0	1123.27	1170.0	1145.0	1195.0

247 rows × 6 columns

Gambar 3.12. Data Validasi PGAS

Validation Data PTBA.JK						
	Closing	WEMA 5	WEMA 50	Open	Low	High
Date						
2023-01-24 00:00:00+07:00	3540.0	3540.0	3654.62	3560.0	3510.0	3560.0
2023-01-25 00:00:00+07:00	3480.0	3480.0	3657.58	3540.0	3480.0	3540.0
2023-01-26 00:00:00+07:00	3390.0	3390.0	3658.59	3440.0	3360.0	3440.0
2023-01-27 00:00:00+07:00	3470.0	3470.0	3666.19	3430.0	3420.0	3480.0
2023-01-30 00:00:00+07:00	3450.0	3450.0	3667.92	3470.0	3430.0	3540.0
...	...	...	...	...	...	...
2024-01-26 00:00:00+07:00	2570.0	2570.0	2437.97	2600.0	2540.0	2600.0
2024-01-29 00:00:00+07:00	2570.0	2570.0	2441.39	2570.0	2560.0	2590.0
2024-01-30 00:00:00+07:00	2590.0	2590.0	2445.35	2560.0	2540.0	2610.0
2024-01-31 00:00:00+07:00	2610.0	2610.0	2450.22	2610.0	2600.0	2660.0
2024-02-01 00:00:00+07:00	2560.0	2560.0	2451.74	2610.0	2560.0	2610.0

247 rows × 6 columns

Gambar 3.13. Data Validasi PTBA

Terakhir, data *testing* terdiri dari 10 persen dataset atau 248 baris digunakan untuk mengukur kemampuan model untuk memprediksi data yang belum pernah dilihat sebelumnya. Data *testing* dimulai dari tanggal dua Februari 2024 sampai dengan 21 Februari 2025. Data *testing* dapat dilihat pada Gambar 3.14, 3.15, dan 3.16.



Testing Data ADRO.JK						
	Closing	WEMA 5	WEMA 50	Open	Low	High
Date						
2024-02-02 00:00:00+07:00	2420.0	2401.78	2525.72	2420.0	2390.0	2440.0
2024-02-05 00:00:00+07:00	2430.0	2410.89	2521.65	2420.0	2420.0	2470.0
2024-02-06 00:00:00+07:00	2460.0	2423.56	2518.99	2450.0	2440.0	2470.0
2024-02-07 00:00:00+07:00	2460.0	2431.56	2516.59	2460.0	2440.0	2470.0
2024-02-12 00:00:00+07:00	2470.0	2443.33	2514.90	2460.0	2450.0	2490.0
...	...	...	...	...	...	...
2025-02-17 00:00:00+07:00	2280.0	2228.00	2443.17	2240.0	2210.0	2300.0
2025-02-18 00:00:00+07:00	2260.0	2224.44	2428.56	2280.0	2250.0	2300.0
2025-02-19 00:00:00+07:00	2250.0	2242.44	2440.09	2270.0	2210.0	2290.0
2025-02-20 00:00:00+07:00	2200.0	2227.56	2441.35	2250.0	2190.0	2250.0
2025-02-21 00:00:00+07:00	2180.0	2228.89	2441.78	2200.0	2180.0	2240.0

248 rows × 6 columns

Gambar 3.14. Data *Testing* ADRO

Testing Data PGAS.JK						
	Closing	WEMA 5	WEMA 50	Open	Low	High
Date						
2024-02-02 00:00:00+07:00	1135.0	1134.11	1121.53	1160.0	1130.0	1165.0
2024-02-05 00:00:00+07:00	1140.0	1138.22	1121.16	1140.0	1130.0	1155.0
2024-02-06 00:00:00+07:00	1155.0	1146.11	1121.37	1145.0	1145.0	1165.0
2024-02-07 00:00:00+07:00	1155.0	1154.11	1120.63	1155.0	1145.0	1160.0
2024-02-12 00:00:00+07:00	1160.0	1152.22	1120.28	1155.0	1140.0	1165.0
...	...	...	...	...	...	...
2025-02-17 00:00:00+07:00	1685.0	1642.11	1602.44	1650.0	1635.0	1690.0
2025-02-18 00:00:00+07:00	1700.0	1658.22	1605.73	1685.0	1685.0	1715.0
2025-02-19 00:00:00+07:00	1685.0	1659.67	1608.72	1710.0	1670.0	1710.0
2025-02-20 00:00:00+07:00	1685.0	1665.89	1610.51	1685.0	1675.0	1700.0
2025-02-21 00:00:00+07:00	1660.0	1669.78	1610.82	1685.0	1645.0	1685.0

248 rows × 6 columns

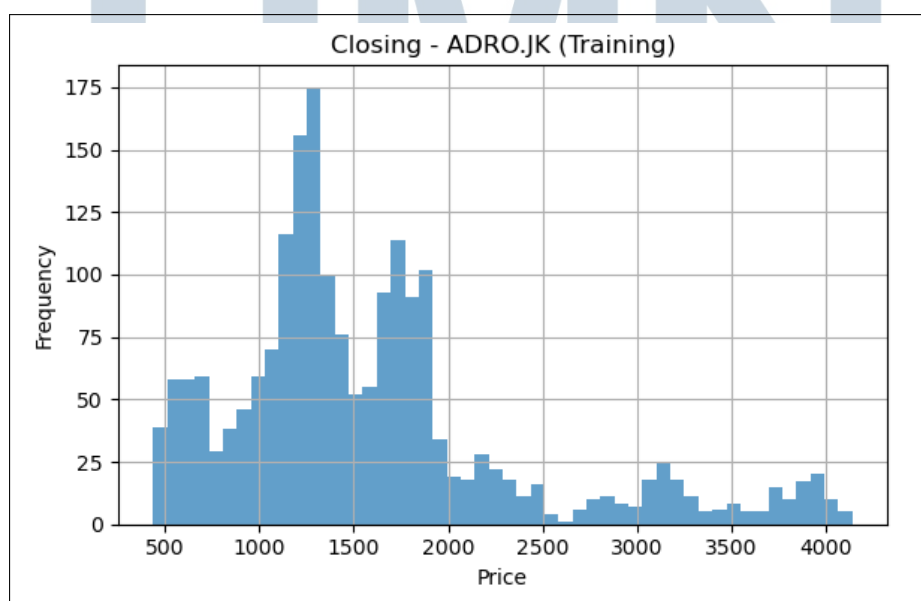
Gambar 3.15. Data *Testing* PGAS

Testing Data PTBA.JK						
	Closing	WEMA 5	WEMA 50	Open	Low	High
Date						
2024-02-02 00:00:00+07:00	2540.0	2565.78	2453.80	2560.0	2540.0	2580.0
2024-02-05 00:00:00+07:00	2540.0	2566.67	2456.35	2550.0	2540.0	2590.0
2024-02-06 00:00:00+07:00	2560.0	2572.89	2460.15	2550.0	2540.0	2570.0
2024-02-07 00:00:00+07:00	2610.0	2583.33	2464.84	2560.0	2560.0	2610.0
2024-02-12 00:00:00+07:00	2650.0	2586.00	2468.13	2610.0	2610.0	2660.0
...	...	...	...	...	...	...
2025-02-17 00:00:00+07:00	2710.0	2662.89	2696.19	2710.0	2680.0	2710.0
2025-02-18 00:00:00+07:00	2690.0	2668.22	2691.36	2710.0	2680.0	2720.0
2025-02-19 00:00:00+07:00	2660.0	2673.33	2689.52	2700.0	2660.0	2720.0
2025-02-20 00:00:00+07:00	2670.0	2688.22	2690.38	2670.0	2630.0	2700.0
2025-02-21 00:00:00+07:00	2660.0	2684.44	2689.29	2660.0	2640.0	2690.0

248 rows × 6 columns

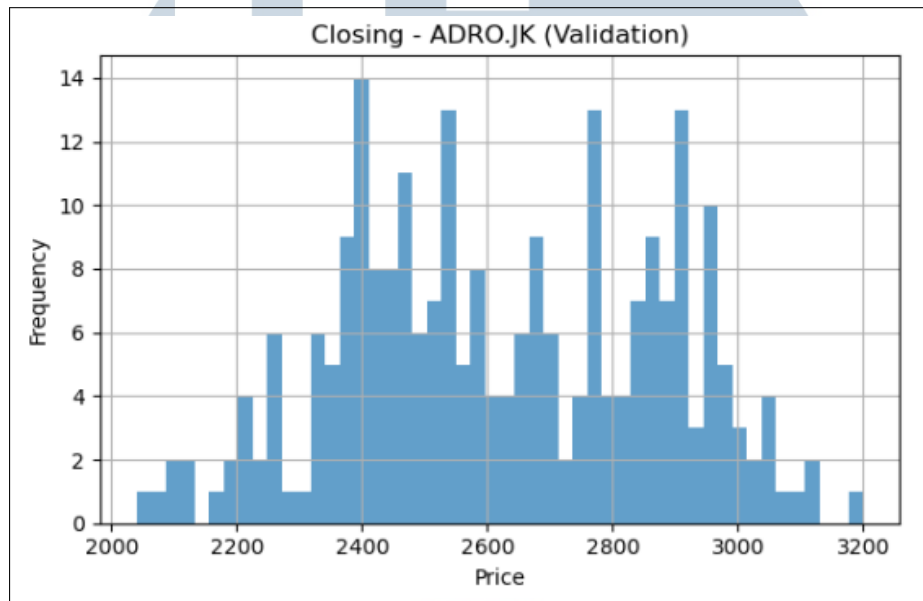
Gambar 3.16. Data *Testing* PTBA

Pada Gambar 3.17 dapat dilihat penyebaran dari data *training* untuk emiten ADRO. Rentang harga penutupan saham berada di sekitar 400an sampai dengan 4200an. Jumlah data paling banyak berada pada harga 1000 sampai 1800. Penyebaran saham lebih banyak berada di sebelah kiri atau harga kurang dari 2000. Jumlah harga yang bernilai lebih dari 3000 jauh lebih sedikit sehingga model kemungkinan kurang dalam mempelajari harga dengan rentang tersebut.



Gambar 3.17. Distribusi *training* ADRO

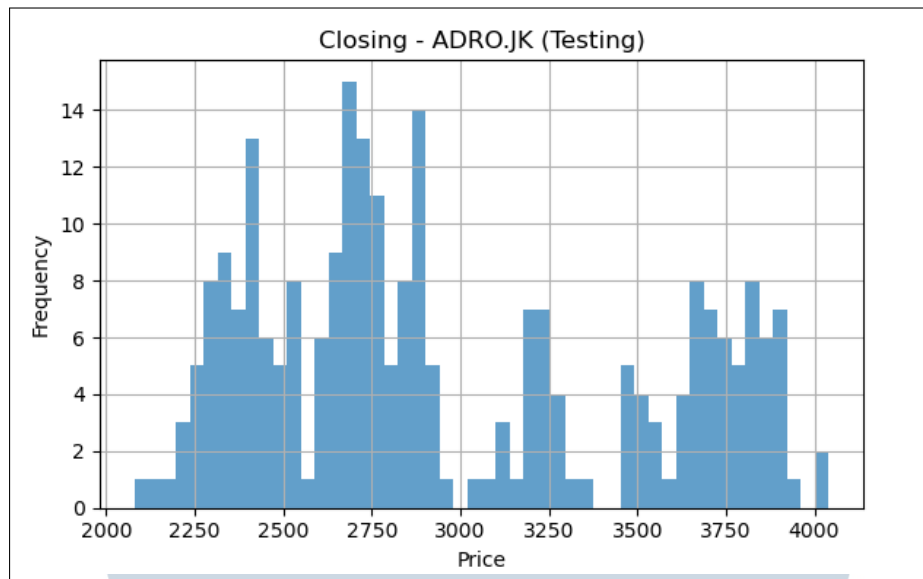
Penyebaran data *validation* ADRO dapat dilihat pada Gambar 3.18. Tidak seperti data *training*, penyebaran data *validation* terlihat hampir simetris. Harga saham paling banyak berada dalam rentang 2000 sampai 3000. Terdapat satu nilai ekstrim pada harga 3200. Penyebaran ini relatif lebih mudah dibanding dengan data *training*.



Gambar 3.18. Distribusi *validation* ADRO

Berikutnya terdapat penyebaran dari data *testing* ADRO pada Gambar 3.19 yang dimulai dari tanggal dua Februari sampai dengan 21 Februari 2025. Rentang harga berada dari harga 2200an sampai 4000an. Tingkat fluktuatif menjadi sangat tinggi dengan penyebaran data yang tidak terkonsentrasi pada kisaran harga tertentu. Hal ini mungkin dipengaruhi oleh faktor eksternal seperti sentimen pasar, aksi korporasi besar yang dilakukan ADRO, dan sebagainya. Dengan ketidakstabilan ini kemungkinan model sulit untuk memprediksi dengan baik pada saat evaluasi akhir model menggunakan RMSE dan MAPE.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.19. Distribusi *testing* ADRO

Kode proses pembagian dataset dapat dilihat pada Kode 3.1. Pada kode tersebut untuk mendapat ukuran data *training*, panjang dari dataset dikalikan dengan 0.80. Selanjutnya ukuran data validasi dihitung dari panjang dataset yang dikalikan dengan 0.10. Data *training* didapat dari *dataset* indeks 0 sampai dengan indeks ukuran data *training*, data validasi dari indeks ukuran data *training* sampai *val_end* yang dihitung dengan menjumlahkan *train_size* dengan *val_size*, terakhir data *testing* didapat dari dataset indeks *val_end* sampai indeks terakhir.

```

1 train_size = {}
2 val_size = {}
3 val_end = {}
4 train_data = {}
5 test_data = {}
6 val_data = {}
7
8 for ticker in tickers:
9     print(len(df[ticker]))
10    train_size[ticker] = int(len(df[ticker]) * 0.80)
11    val_size[ticker] = int(len(df[ticker]) * 0.10)
12
13    val_end[ticker] = train_size[ticker] + val_size[ticker]
14
15    train_data[ticker] = df[ticker].iloc[:train_size[ticker]]
16    val_data[ticker] = df[ticker].iloc[train_size[ticker]:val_end[
17    ticker]]
18    test_data[ticker] = df[ticker].iloc[val_end[ticker]:]
19
20    print(f'Training max {ticker}: {train_data[ticker].max()}')
21    print(f'Training length: {len(train_data[ticker])}')
22
23    print(f'Validation max {ticker}: {val_data[ticker].max()}')
24    print(f'Validation length: {len(val_data[ticker])}')

```

```

25 print(f'Testing max {ticker}: {test_data[ticker].max()}')
26 print(f'Testing length: {len(test_data[ticker])}')

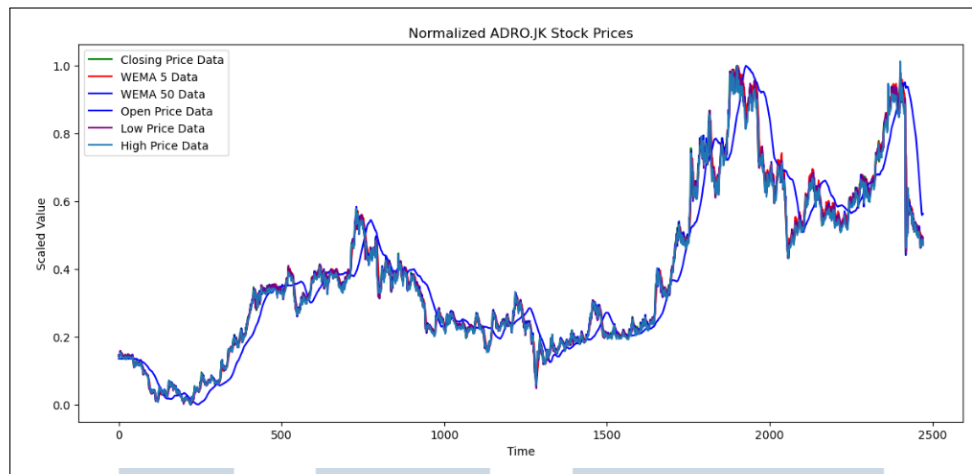
```

Kode 3.1: Pembagian dataset menjadi *training*, *validation*, dan *testing*

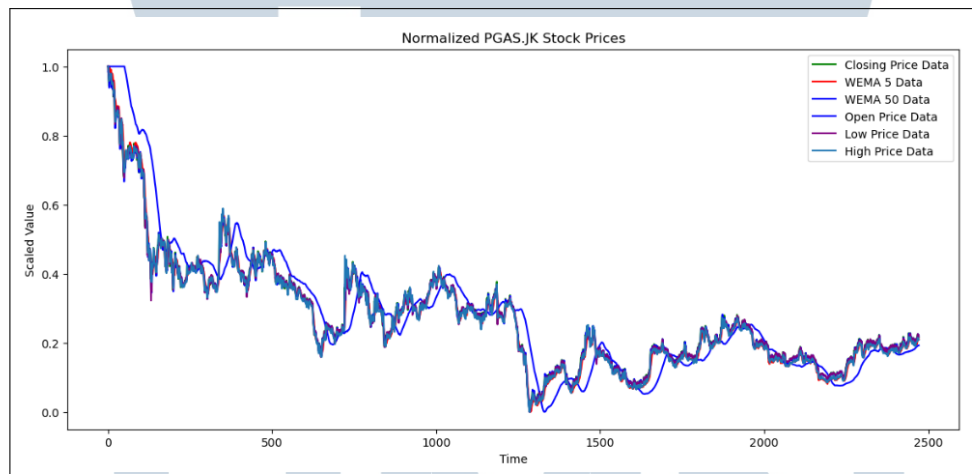
3.9.3 Normalisasi Data

Normalisasi dilakukan sebelum data diproses untuk menyamakan format data menjadi rentang nol sampai satu. Tujuannya untuk mempermudah fungsi aktivasi tanh dan sigmoid pada LSTM untuk memproses data. Pada satu kasus, misal input sangat besar seperti 1000 atau 2000 saat memasuki fungsi aktivasi tanh dan sigmoid maka keluarannya tidak dapat dibedakan satu sama lain atau mengalami saturasi. Hal ini menyebabkan kemampuan belajar dari model menjadi kurang efektif [31]. Oleh karena itu, dengan melakukan normalisasi terlebih dahulu sebelum data diproses, nilai input dapat berada di rentang yang tidak terlalu ekstrim sehingga penggunaan fungsi aktivasi tanh dan sigmoid menjadi lebih efektif karena keluaran dari fungsi aktivasi tidak mengalami saturasi. Min max juga dapat membuat model lebih cepat stabil dan konvergen [32]. Berdasarkan penelitian sebelumnya, *min max normalization* dapat meningkatkan performa model dan mendapat nilai evaluasi RMSE dan MAPE yang rendah [33]. Pada penelitian tersebut juga disampaikan bahwa penggunaan min max normalization pada LSTM lebih unggul dari penggunaan metode normalisasi lain seperti Z Score. Normalisasi juga harus dilakukan untuk mencegah adanya dominasi fitur yang mempengaruhi proses prediksi. Apalagi dengan LSTM yang menggunakan banyak fitur dalam memprediksi harga saham seperti yang digunakan pada penelitian ini yang mempunyai rentang harga yang berbeda-beda. Grafik dari data yang telah dinormalisasi dapat dilihat pada Gambar 3.20, 3.21, dan 3.22.

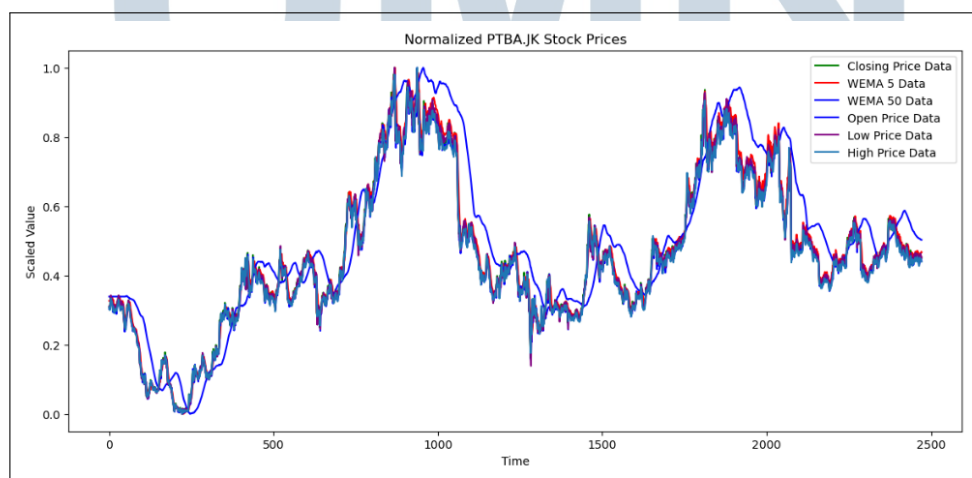
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.20. Grafik Normalisasi ADRO



Gambar 3.21. Grafik Normalisasi PGAS



Gambar 3.22. Grafik Normalisasi PTBA

3.9.4 Reshape Data

Tahapan terakhir sebelum data digunakan pada model LSTM adalah mengubah bentuk data menjadi data tiga dimensi. LSTM membutuhkan bentuk data dalam bentuk *timeseries* sehingga sebelum diproses data harus diubah ke bentuk tiga dimensi [34]. Bentuk tiga dimensi ini harus digunakan untuk dapat diproses LSTM dari *library Keras Python* [8]. Bentuk tiga dimensi tersebut berisi jumlah baris, jumlah hari yang digunakan untuk memprediksi, dan jumlah fitur yang digunakan [35]. Hubungan antar fitur dengan jumlah hari dapat mempengaruhi kinerja model. Panjang hari yang digunakan untuk memprediksi sangat penting untuk menggambarkan tren dari data sebelumnya [36].

Penelitian ini menggunakan model LSTM yang memprediksi harga saham berdasarkan dari 60 hari sebelumnya. Penggunaan 60 hari diambil dari penelitian [10]. Kode untuk mengubah data menjadi bentuk tiga dimensi dapat dilihat pada Kode 3.2. Data yang telah dinormalisasi dimasukkan ke fungsi `create_sequence` untuk mempersiapkan data dalam bentuk *time series*. Fungsi tersebut bekerja dengan menerima dua parameter yaitu data yang berisi data yang telah dinormalisasi sebelumnya dan `seq_length` atau jumlah hari yang digunakan untuk memprediksi harga saham yang dalam penelitian ini digunakan 60 hari. Fungsi ini mengeluarkan dua baris data yaitu X dan y. X berisi potongan data sepanjang `seq_length` yang digunakan sebagai input pada model LSTM dan y berisi nilai asli dari harga yang diprediksi.

Data *training* yang dibagi jadi X dan y dalam penelitian ini disebut `X_train` dan `y_train`, data *validation* disebut `X_val` dan `y_val`, dan data *testing* disebut `X_test` dan `y_test`. Masing-masing bentuk tiga dimensi data X adalah (1917, 60, 6), (187, 60, 6), dan (188, 60, 6).

```
1 def create_sequence(data, seq_length):
2     X = []
3     y = []
4     for i in range(len(data) - seq_length):
5         X.append(data[i:i+seq_length])
6         y.append(data[i+seq_length][0])
7     return np.array(X), np.array(y)
8
9 seq_length = 60
10
11 X_train = {}
12 y_train = {}
13
14 X_val = {}
15 y_val = {}
16
17 X_test = {}
```

```

18 y_test = {}
19
20
21 for ticker in tickers:
22     X_train[ticker], y_train[ticker] = create_sequence(
23         train_scaled[ticker], seq_length)
24     X_test[ticker], y_test[ticker] = create_sequence(test_scaled[
25         ticker], seq_length)
26     X_val[ticker], y_val[ticker] = create_sequence(val_scaled[
27         ticker], seq_length)
28
29     print('Training Shape')
30     print(X_train[ticker].shape)
31     print(y_train[ticker].shape)
32
33     print('Validation Shape')
34     print(X_val[ticker].shape)
35     print(y_val[ticker].shape)
36
37     print('Testing Shape')
38     print(X_test[ticker].shape)
39     print(y_test[ticker].shape)

```

Kode 3.2: Melakukan *Reshape data* untuk menyesuaikan dengan format model *Long Short-Term Memory*

3.10 Pembuatan Model

Penelitian ini menggunakan model LSTM dengan lima *layers* yang terdiri dari dua lapisan LSTM, satu lapisan *dropout*, dan dua lapisan *Dense*. Metode *tuning hyperparameter* dilakukan dengan metode *Grid Search*. *Grid search* merupakan metode optimasi *hyperparameter* pada model *deep learning*. Metode ini bekerja dengan membuat percobaan penggunaan parameter untuk mencari parameter yang paling optimal pada model berjalan secara otomatis [26]. Proses *Grid Search* dapat dilihat pada Gambar 3.3.

Sebelum dilakukan *Hyperparameter tuning* untuk mencari *hyperparameter* terbaik, model ini menggunakan *hyperparameter* dasar *learning rate* sebesar 0.001, *optimizer* dengan menggunakan *Adam Optimizer*, *units LSTM* sebesar 100, *dropout rate* 0.4, jumlah *epoch* adalah 100, dan *batch size* sebesar 32. Ketika proses pelatihan model dilakukan, fungsi *EarlyStopping* dari *library Python Tensorflow* digunakan untuk menghentikan pelatihan jika selama 10 iterasi atau *epoch* nilai *validation loss* stagnan atau bahkan performa mengalami penurunan. *Mean squared error* digunakan sebagai evaluasi *loss* saat pelatihan. Pada tahap evaluasi penelitian ini, nilai MAPE dan RMSE sebelum dan setelah *tuning hyperparameter* dibandingkan untuk mengetahui seberapa besar pengaruh dari *tuning hyperparameter* terhadap kinerja model.

Proses pelatihan menggunakan model LSTM yang dibuat dengan fungsi `create_lstm_model`. Fungsi tersebut mempunyai parameter bentuk atau *shape* dari data *input* `X_train` dan *hyperparameter* yang telah dioptimasi. Proses pelatihan dihentikan menggunakan library `EarlyStopping` untuk mencegah terjadinya *overfitting* pada model. Untuk validasi data, model menggunakan data `X_val` dan `y_val`. Model yang telah dilatih disimpan pada variabel `history`.

```

1 for ticker in tickers:
2     print(f'Predict Stock {ticker} With Hyperparameter Tuning')
3     set_seed(15)
4     model_with_tuning[ticker] = create_lstm_model(
5         (X_train[ticker].shape[1], X_train[ticker].shape[2]),
6         units=hyperparams[ticker]['units'],
7         dropout_rate=hyperparams[ticker]['dropout_rate'],
8         optimizer=hyperparams[ticker]['optimizer'],
9         learning_rate=hyperparams[ticker]['learning_rate']
10    )
11
12    early_stop = EarlyStopping(monitor='val_loss', patience=10,
13                               restore_best_weights=True)
14
15    history[ticker] = model_with_tuning[ticker].fit(
16        X_train[ticker], y_train[ticker],
17        validation_data=(X_val[ticker], y_val[ticker]),
18        epochs=hyperparams[ticker]['epochs'],
19        batch_size=hyperparams[ticker]['batch_size'],
20        callbacks=[early_stop],
21        verbose=1,
22        shuffle=False
    )

```

Kode 3.3: Melakukan pelatihan dengan model *Long Short-Term Memory* penelitian ini

3.11 Denormalisasi Data

Sebelum nilai harga prediksi saham dievaluasi, nilai yang telah dinormalisasi pada tahap pra proses data harus dikembalikan ke format semula. Ini dilakukan dengan denormalisasi data seperti pada rumus 2.8. Pada Kode 3.4 proses denormalisasi dilakukan dengan fungsi `close_scaler.inverse_transform`. Fungsi ini secara otomatis akan melakukan denormalisasi terhadap data normalisasi yang dimasukkan pada parameter fungsi. Pada kasus ini data yang dimasukkan ke fungsi adalah data prediksi *testing*, data prediksi *training*, data *testing*, dan data *training*.

```

1 y_pred_inv = close_scaler[ticker].inverse_transform(y_pred)
2 y_train_pred_inv = close_scaler[ticker].inverse_transform(
3     y_train_pred)
4 y_test_inv = close_scaler[ticker].inverse_transform(y_test[ticker]
5     ].reshape(-1, 1))

```

```
s y_train_inv = close_scaler[ticker].inverse_transform(y_train[
    ticker].reshape(-1, 1))
```

Kode 3.4: Melakukan denormalisasi untuk mengembalikan format data

3.12 Evaluasi Model

Pada tahap ini dilakukan pengujian pada model LSTM yang digunakan. Pengujian dilakukan dengan menggunakan metrik evaluasi MAPE dan RMSE dengan membandingkan nilai asli dengan nilai prediksi harga saham.

