

PREDIKSI HARGA BERAS MENGGUNAKAN JARINGAN SYARAF TIRUAN DENGAN ALGORITMA LONG - SHORT TERM MEMORY

Mohammad Nurul Ahid^{1*}, Imadudin Harjanto², Muhammad Amiruddin³

¹²³ Universitas PGRI Semarang

Jl. Lontar No. 24, Karangtempel, Kec. Semarang Timur, Kota Semarang, Jawa Tengah, 50232, Indonesia

*Email: ahid764mna@gmail.com

Abstrak— Beras merupakan komoditas pangan utama masyarakat Indonesia dengan beragam mutu, mulai dari premium hingga medium. Perbedaan kualitas beras memengaruhi harga jual, di mana beras premium dijual lebih mahal karena dianggap bermutu tinggi, sedangkan beras medium lebih terjangkau dan umum dikonsumsi. Harga beras sering mengalami fluktuasi akibat berbagai faktor seperti kebijakan pemerintah, produksi pertanian, permintaan konsumen, dan ketersediaan stok. Penelitian ini bertujuan memprediksi harga beras medium di tingkat penggilingan menggunakan metode Jaringan Syaraf Tiruan (Artificial Neural Network) dengan algoritma Long Short-Term Memory (LSTM). Data yang digunakan berupa harga bulanan beras medium periode 2014–2024 dari Badan Pusat Statistik. Proses penelitian meliputi preprocessing data, pembuatan jendela waktu dengan Fast Fourier Transform (FFT), data splitting, pembangunan arsitektur LSTM, pelatihan model, evaluasi, dan visualisasi prediksi dengan variasi window size 12, 24, dan 36. Hasil penelitian menunjukkan bahwa model LSTM memiliki performa terbaik pada window size 12 dengan nilai Mean Absolute Percentage Error (MAPE) sebesar 6,58%, dibandingkan 8,53% pada window size 24 dan 10,13% pada window size 36. Meskipun persentase error yang dihasilkan masih relatif tinggi, model LSTM terbukti efektif dalam memprediksi harga beras dengan hasil persentase error terkecil yaitu 6,58% dari jumlah window size 12. Jumlah error yang dihasilkan dapat diperkecil lagi dengan menambahkan data lain yang relevan misal faktor cuaca musiman, hingga kebijakan pemerintah sebagai fitur input. Dengan demikian, meskipun masih terdapat error, model LSTM terbukti efektif dalam memprediksi harga beras dan dapat ditingkatkan akurasi dengan menambahkan variabel relevan seperti faktor musiman dan kebijakan pemerintah.

Kata kunci: Flutuasi; Jaringan Syarat Tiruan; LSTM; Mean Absolute Percentage Error

I. PENDAHULUAN

Beras merupakan makanan pokok masyarakat Indonesia dengan variasi mutu, seperti premium, medium, hingga submedium. Harga beras sering berfluktuasi karena dipengaruhi kebijakan pemerintah, produksi, permintaan, hingga distribusi. Fluktuasi ini dapat berdampak pada stabilitas ekonomi masyarakat. Oleh karena itu, diperlukan metode prediksi harga yang akurat untuk membantu pengambilan kebijakan. Penelitian ini menggunakan metode Long Short-Term Memory (LSTM), salah satu algoritma Recurrent Neural Network yang mampu menangani data sekuensial dan mendeteksi pola jangka panjang.

Dengan adanya perbedaan harga di setiap kualitas atau klasifikasi jenis beras, serta faktor - faktor lain seperti kebijakan pemerintah, produksi pertanian, permintaan konsumen, persediaan beras, dan perubahan gaya hidup, maka harga beras di pasaran tentunya kurang stabil atau sering mengalami fluktuasi harga, sehingga dengan ini, penulis melakukan penelitian untuk memberikan prediksi harga beras di tingkat penggilingan dengan kualitas medium. Dalam penelitian ini, menggunakan data rata - rata harga beras medium di tingkat penggilingan dari tahun 2014 sampai 2024. Pada data tersebut terdapat fluktuasi harga yang signifikan.

II. STUDI PUSTAKA

Pada penelitian yang dilakukan oleh Cahyani, J., Mujahidin, S., & Fiqar, T. P. (2023) dengan judul Implementasi Metode Long Short – Term Memory (LSTM) untuk Memprediksi Harga Bahan Pokok Nasional, data mining merupakan suatu proses untuk menemukan pola atau pengetahuan tersembunyi dari sejumlah besar data. Penerapan data mining dalam konteks harga beras memungkinkan ditemukannya pola historis harga yang dapat digunakan untuk memperkirakan harga pada periode berikutnya. Seiring perkembangan teknologi, metode prediksi berbasis machine learning semakin banyak digunakan karena kemampuannya dalam mempelajari data historis dan menghasilkan model prediksi yang lebih akurat dibandingkan metode konvensional.

Hasil penelitian dari (Ilmananda & Alfianty, 2022), dengan judul “Prediksi Rata-Rata Harga Beras Nasional Dengan Kualitas Premium di Tingkat Penggilingan Hingga Akhir Tahun 2022” Data yang digunakan pada penelitian ini yaitu data dari tahun 2018 sampai tahun 2022 yang bersumber dari Badan Pusat Statistik (BPS) Indonesia. Penelitian ini dilakukan dengan pendekatan kuantitatif dengan metode double exponential smoothing dan ukuran kesalahan dalam penelitian tersebut menggunakan mean absolute percentage error (mape).

III. METODE/DESAIN

Prosedur Penelitian

Prosedur penelitian dalam skripsi ini adalah sebagai berikut:

1. Mengumpulkan dataset berupa harga beras di Indonesia dengan kualitas premium dan medium periode tahun 2024 sampai dengan 2025 yang akan disajikan dalam bentuk data bulanan. Data yang digunakan merupakan data sekunder yang diperoleh dari website Badan Pusat Statistik (BPS)
2. Melakukan preprocessing data, yaitu proses menyiapkan data sebelum disajikan dalam model peramalan Long – Short Term Memory. Langkah – langkahnya, antaralain:
 - a. Pembersihan data (data cleaning), meliputi penanganan data yang hilang (missing value) menggunakan metode interpolasi. Menggunakan metode interpolasi untuk menangani missing values adalah langkah penting dalam analisis data. Dengan memilih metode yang tepat, dapat meningkatkan kualitas dataset dan hasil analisis.
 - b. Normalisasi data, menggunakan Min-Max Scalling dari tools Skicit-learn. Memilih metode normalisasi yang tepat sangat penting untuk meningkatkan kualitas analisis data dan hasil model. Normalisasi membantu dalam menciptakan dataset yang lebih seimbang dan dapat diandalkan untuk analisis lebih lanjut.
 - c. Teknik preprocessing data ini menggunakan tools dari Google Colaboratory dan menggunakan Bahasa pemrograman Phyton dengan library Pandas, dan NumPy.
3. Melakukan training model, yaitu membagi data menjadi data training dan data testing melalui tahap – tahap sebagai berikut:
4. Pembagian data menjadi data training dan data testing adalah langkah penting dalam proses pengembangan model machine learning. Data training digunakan untuk melatih model, sedangkan data testing digunakan untuk mengevaluasi kinerja model. Di Python, dapat menggunakan pustaka scikit-learn untuk melakukan pembagian data ini dengan mudah.
5. Menentukan fungsi loss, merupakan alat penting dalam machine learning yang digunakan untuk mengukur seberapa baik model memprediksi hasil yang diinginkan. Dalam penelitian ini menggunakan fungsi loss Mean Absolute Percentage Error (MAPE).
6. Menghitung jumlah epoch dan batch size, merupakan dua parameter penting yang perlu dipertimbangkan. Keduanya berpengaruh pada cara model belajar dari data dan dapat mempengaruhi kinerja serta waktu pelatihan. Dalam proses ini menggunakan tools dari phyton yaitu Keras.
7. Melakukan testing model, yaitu proses evaluasi kinerja model setelah dilatih menggunakan data pelatihan. Tujuan utama dari pengujian model adalah untuk menilai seberapa baik model dapat menggeneralisasi pada data yang belum pernah dilihat sebelumnya, yaitu data testing. Langkah ini menggunakan metrik evaluasi Mean Absolute Percentage Error (MAPE)) dengan tools

pustaka scikit-learn dari Phyton. Testing model adalah langkah krusial dalam pengembangan model machine learning. Dengan melakukan pengujian, dapat dipastikan bahwa model tidak hanya belajar dari data training, tetapi juga dapat memberikan prediksi yang akurat pada data baru.

8. Menganalisa tingkat akurasi model arsitektur

Langkah ini meliputi :

- a. Membandingkan nilai prediksi dan nilai aktual.
- b. Interpretasi metrik evaluasi Mean Absolute Error (MAPE).
- c. Analisis tren prediksi model dibandingkan dengan tren actual meliputi visualisasi dan analisis menggunakan tools Matplotlib di Phyton.

IV. HASIL DAN PEMBAHASAN

Menyiapkan Data

Pada penelitian ini menggunakan data rata – rata harga beras bulanan di tingkat penggilingan menurut kualitas antara tahun 2014 – 2024. Terdapat 132 baris data (132 bulan) pada data tersebut. Data harga beras berbentuk file excel yang telah dimasukkan akan dibaca menggunakan library NumPy. Kemudian, mengambil kolom 'Medium' sebagai data harga yang akan diolah, lalu mengubah data dalam kolom tersebut menjadi array NumPy, dan menangani nilai yang hilang (jika ada) dengan interpolasi. Data harga kemudian dinormalisasi menggunakan MinMaxScaler agar nilainya berada dalam rentang [0, 1] untuk stabilitas pelatihan model.

Membuat jendela waktu + fft

Fungsi make window with fft dibuat untuk memproses data deret waktu menjadi jendela-jendela (sequences) untuk input LSTM dan mengekstrak fitur frekuensi menggunakan Fast Fourier Transform (FFT) untuk input Dense. Setiap jendela memiliki ukuran WINDOW (14 bulan) dan digeser sebanyak STEP (1 bulan). Outputnya adaah X time (jendela waktu), X freg (magnitude FFT), dan y (harga bulan berikutnya sebagai target).

Window size merupakan panjang jendela historis atau jumlah bulan ke belakang yang digunakan model LSTM untuk melakukan prediksi harga bebas di masa mendatang. Jika window terlalu pendek, model hanya melihat sedikit pola dan kemungkinan tidak cukup untuk menangkap tren musiman (tahunan). Sedangkan jika window size yang terlalu panjang, model dapat melihat pola lebih jauh namun bisa juga terjadi overfitting sehingga informasi tidak relevan noise.

Data splitting

Data yang sudah di-window dibagi menjadi data latih (train) dan data uji (test). Pembagian dilakukan secara berurutan (tanpa shuffle) karena ini adalah data deret waktu, dengan 80% untuk data latih dan 20% untuk data uji.

Membuat model arsitektur LSTM

Model neural network dibangun menggunakan Keras API dari TensorFlow. Model ini memiliki dua input: satu untuk data deret waktu (inp_time) yang diproses oleh layer LSTM dan Dropout, dan satu lagi untuk fitur frekuensi

(inp_freq) yang diproses oleh layer Dense dan Dropout. Output dari kedua cabang ini digabungkan (Concatenate) dan kemudian dilewatkan melalui layer Dense tambahan sebelum layer output final (Dense(1)) yang memprediksi harga bulan berikutnya (dalam skala [0, 1]). Model dikompilasi dengan optimizer Adam dan fungsi loss Mean Absolute Percentage Error(MAPE).

Sehingga diperoleh model seperti berikut :

1. Model arsitektur LSTM dari window size = 24

Layer (type)	Output Shape	Param #	Connected to
time_window (InputLayer)	(None, 24, 1)	0	-
freq_features (InputLayer)	(None, 12)	0	-
lstm_2 (LSTM)	(None, 64)	16,896	time_window[0][0]
dense_6 (Dense)	(None, 32)	416	freq_features[0]...
dropout_4 (Dropout)	(None, 64)	0	lstm_2[0][0]
dropout_5 (Dropout)	(None, 32)	0	dense_6[0][0]
concatenate_2 (Concatenate)	(None, 96)	0	dropout_4[0][0], dropout_5[0][0]
dense_7 (Dense)	(None, 32)	3,104	concatenate_2[0]...
dense_8 (Dense)	(None, 1)	33	dense_7[0][0]

Total params: 20,449 (79.88 KB)
Trainable params: 20,449 (79.88 KB)
Non-trainable params: 0 (0.00 B)

Gambar 1. Model layer window size 24

Pada gambar 1 menampilkan neural network yang telah dibuat, terdapat total parameter sebanyak 20.499 dari tiap layer yang terdiri dari 16.896 parameter dari model lstm, 2.416 parameter dari model dense_6, 3.104 parameter dari model dense_7, dan 33 parameter dari model dense_8.

2. Model arsitektur LSTM dari windows size = 36

Layer (type)	Output Shape	Param #	Connected to
time_window (InputLayer)	(None, 36, 1)	0	-
freq_features (InputLayer)	(None, 18)	0	-
lstm (LSTM)	(None, 64)	16,896	time_window[0][0]
dense (Dense)	(None, 32)	608	freq_features[0]...
dropout (Dropout)	(None, 64)	0	lstm[0][0]
dropout_1 (Dropout)	(None, 32)	0	dense[0][0]
concatenate (Concatenate)	(None, 96)	0	dropout[0][0], dropout_1[0][0]
dense_1 (Dense)	(None, 32)	3,104	concatenate[0][0]
dense_2 (Dense)	(None, 1)	33	dense_1[0][0]

Total params: 20,641 (80.63 KB)
Trainable params: 20,641 (80.63 KB)
Non-trainable params: 0 (0.00 B)

Gambar 2. Model layer window size 36

Pada Gambar 2. menampilkan neural network yang telah dibuat, terdapat total parameter sebanyak 20.641 dari tiap layer yang terdiri dari 16.896 parameter dari model lstm, 608 parameter dari model dense, 3.104 parameter dari model dense_1, dan 33 parameter dari model dense_2

3. Model arsitektur LSTM dari window size = 12

Layer (type)	Output Shape	Param #	Connected to
time_window (InputLayer)	(None, 12, 1)	0	-
freq_features (InputLayer)	(None, 6)	0	-
lstm (LSTM)	(None, 64)	16,896	time_window[0][0]
dense (Dense)	(None, 32)	224	freq_features[0]...
dropout (Dropout)	(None, 64)	0	lstm[0][0]
dropout_1 (Dropout)	(None, 32)	0	dense[0][0]
concatenate (Concatenate)	(None, 96)	0	dropout[0][0], dropout_1[0][0]
dense_1 (Dense)	(None, 32)	3,104	concatenate[0][0]
dense_2 (Dense)	(None, 1)	33	dense_1[0][0]

Total params: 20,257 (79.13 KB)
Trainable params: 20,257 (79.13 KB)
Non-trainable params: 0 (0.00 B)

Gambar 3. Model layer window size 12

Pada Gambar 3 menampilkan neural network yang telah dibuat, terdapat total parameter sebanyak 20.257 dari tiap layer yang terdiri dari 16.896 parameter dari model lstm, 224 parameter dari model dense, 3.104 parameter dari model dense_1, dan 33 parameter dari model dense_2.

Training model

Model dilatih menggunakan data latih (X_time_train, X_freq_train) dan target y_train. Data latih dibagi lagi 10% untuk validasi (validation_split=0.1). EarlyStopping digunakan sebagai callback untuk menghentikan pelatihan jika validasi loss tidak membaik setelah beberapa epoch (patience=20) dan mengembalikan bobot model terbaik. Pelatihan berjalan hingga 100 epoch atau sampai Early Stopping terpicu.

Dalam pelatihan neural network pada penelitian ini, satu periode (epoch) mengacu pada satu lintasan lengkap melalui keseluruhan kumpulan data pelatihan (train). Data pelatihan (train): Model dalam penelitian ini mempelajari data X_time_train dan X_freq_train, dengan y_train sebagai target. Ukuran Batch (batch size): Data pelatihan dibagi menjadi potongan-potongan yang lebih kecil yang disebut batch (dalam penelitian ini, batch_size=16). Model memperbarui bobotnya setelah memproses setiap batch.

Epoch: Ketika model telah melalui semua batch satu kali dalam dataset pelatihan, maka satu epoch telah selesai. Jadi, jika terdapat 100 sampel pelatihan dan ukuran batch 10 maka, satu epoch akan melibatkan 10 pembaruan bobot (100 sampel / 10 ukuran batch = 10 batch). Epochs=100 dalam kode model.fit berarti proses pelatihan akan berulang melalui keseluruhan kumpulan data pelatihan hingga 100 kali. Tujuan pelatihan selama beberapa periode (epoch) adalah supaya model dapat melihat data pelatihan.

Evaluasi dan prediksi model

Setelah pelatihan, model digunakan untuk memprediksi harga pada data uji (X_time_test, X_freq_test). Hasil prediksi (y_pred_scaled) dan data uji aktual (y_test) yang masih dalam skala [0, 1] dikembalikan ke skala asli menggunakan scaler.inverse_transform

Dimana:

- y_test_inv: Ini adalah nilai harga beras aktual pada skala aslinya untuk data test set.

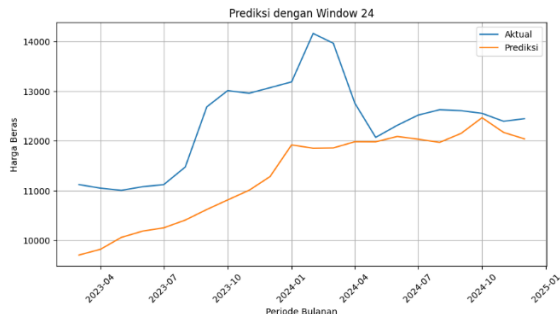
- y_pred_inv : Ini adalah nilai harga beras prediksi oleh model pada skala aslinya untuk data test set.
- $np.abs((y_test_inv - y_pred_inv) / np.clip(y_test_inv, 1e-8, None))$: Ini menghitung persentase error absolut untuk setiap titik data di test set. $np.clip(y_test_inv, 1e-8, None)$ digunakan untuk menghindari pembagian dengan nol jika ada nilai aktual yang sangat mendekati atau sama dengan nol.
- $np.mean(...)$: Ini menghitung rata-rata dari persentase error absolut tersebut di seluruh data test set.
- $* 100$: Hasil rata-rata dikalikan 100 untuk mendapatkan nilai dalam persentase.
- $print(f'MAPE : \{mape:.2f\}\%')$: Baris ini mencetak nilai MAPE yang sudah dihitung, diformat menjadi dua angka desimal.

Jadi, nilai MAPE yang dihasilkan pada output menunjukkan bahwa persentase error (kesalahan) prediksi model menyimpang sekitar 8,58% untuk window size 24, 10,13% untuk window size 36, dan 6.58% untuk window size 12 dari harga aktual pada data test set.

Prediksi bulan berikutnya dari data paling baru

Tahap ini mengambil data dari jendela data paling akhir dari keseluruhan dataset, kemudian melakukan transformasi FFT, dan menggunakan data tersebut sebagai input untuk memprediksi harga beras medium bulan berikutnya dari titik data terakhir yang tersedia. Hasil prediksi dikembalikan ke skala asli dan dicetak.

1. Hasil prediksi window size = 24

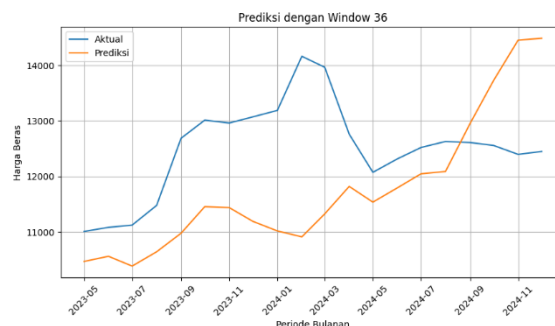


Gambar 4. Grafik hasil prediksi dengan window size 24

Gambar 4 menampilkan grafik– prediksi yang dibuat oleh model menggunakan ukuran jendela 24 bulan. Pada bagian awal periode, yaitu bulan Januari – Juli 2023 harga aktual relatif stabil di kisaran 11.000–11.500. Prediksi juga stabil, tapi mulai dari level lebih rendah (~9700–10300). Artinya model underestimate di awal (nilai prediksi terlalu rendah). Kemudian pada pertengahan periode, yaitu bulan Agustus 2023 – Januari 2024 harga aktual naik tajam (mencapai >14.000). Prediksi juga naik, tetapi jauh lebih lambat dan nilainya tetap lebih rendah. Gap (selisih) antara aktual dan prediksi membesar pada puncak harga (akhir 2023 – awal 2024) yang menunjukkan model jika tidak mampu menangkap lonjakan cepat (lagging). Lalu pada akhir periode, yaitu bulan April – Desember 2024) harga aktual sempat turun tajam ke sekitar 12.000 lalu stabil di 12.500–12.600. Prediksi juga naik kemudian stabil, tetapi nilainya tetap di bawah aktual (sekitar 12.000–

12.400). Pola relatif mirip, tapi model cenderung menghasilkan kurva lebih halus tanpa fluktuasi tajam.

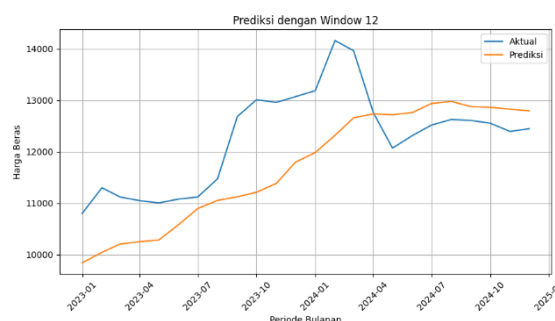
2. Hasil prediksi window size = 36



Gambar 5. Grafik hasil prediksi dengan window size 36

Gambar 5 menampilkan grafik prediksi yang dibuat oleh model menggunakan ukuran jendela 36 bulan. Pada bagian awal, yaitu bulan Mei 2023 – Desember 2024 Periode test yang diprediksi hampir sama dengan window 12, tetapi model melihat riwayat lebih panjang (3 tahun). Kemudian, pada tren awal yaitu bulan Mei – September 2023 harga aktual naik tajam mulai pertengahan 2023. Sedangkan Prediksi tertinggal di bawah aktual (underestimate) dan naik lebih lambat. Artinya model “kurang cepat bereaksi” terhadap kenaikan mendadak. Kemudian pada bagian tengah, yaitu bulan Oktober 2023 – April 2024 harga aktual naik → puncak → lalu turun tajam. Prediksi juga mengikuti pola naik-turun, tapi lebih datar (smoothed). Saat harga aktual melonjak akhir 2023, prediksi jauh di bawah (gap besar). Saat harga aktual turun awal 2024, prediksi tidak turun sedalamnya → overestimate. Lalu pada bagian akhir, yaitu bulan Mei – Desember 2024 harga aktual relatif stabil (sekitar 12.400–12.600). Prediksi justru melonjak sangat tajam setelah Agustus 2024 hingga menembus 14.500. Ini menandakan overfitting atau bias jangka panjang, artinya model memprediksi tren tahunan berlanjut naik, padahal data aktual stabil.

3. Hasil prediksi window size = 12



Gambar 6. Grafik hasil prediksi dengan window size 12

Gambar 6 menampilkan grafik perbandingan antara harga beras aktual dan harga beras prediksi oleh model pada test set, namun kali ini diplot terhadap periode bulanan (tanggal) alih-alih indeks sampel. Pada Bagian Awal (Jan–Sep 2023) Prediksi berada di bawah harga aktual. Artinya model underestimate (harga aktual lebih tinggi dari prediksi). Pola kenaikan lambat masih tertangkap. Kemudian pada bagian Tengah (Okt 2023 – Feb

2024) harga aktual naik tajam sedangkan prediksi juga naik, tetapi lebih halus. Model kurang responsif terhadap lonjakan cepat, sehingga prediksi tertinggal. Lalu, pada bagian akhir (Maret – Des 2024) harga aktual turun tajam lalu stabil. Prediksi juga menurun, tapi cenderung lebih tinggi daripada aktual (overestimate). Model tampak “menahan” pola agar lebih mulus, tidak turun sedalam harga sebenarnya.

Tingkat akurasi kinerja pada model yang telah dibuat kemudian juga diukur (evaluasi) menggunakan metrik evaluasi MAPE (Mean Squared Percentage Error), yaitu mengukur rata rata persentase kesalahan prediksi terhadap nilai aktual. Sehingga menghasilkan jumlah metrik sebagai berikut:

Tabel 1. Hasil Pengukuran MAPE

Window size	Jumlah MAPE
12	6,58%
24	8,53%
36	10.13%

Pada tabel 1 di atas, window 12 (1 tahun ke belakang) menghasilkan nilai MAPE sebesar 6,58% yang artinya prediksi lebih responsif terhadap perubahan harga, sehingga pola prediksi mendekati aktual. Namun, kadang terjadi lag dan over/under shoot pada lonjakan harga. Biasanya MAPE lebih kecil di data yang stabil, lebih besar di saat ada lonjakan tajam. Kemudian pada window 24 (2 tahun ke belakang), menghasilkan nilai MAPE sebesar 8,53 % yang artinya prediksi lebih halus dan stabil, tapi sering underestimate harga aktual. Karena selalu lebih rendah dari aktual, error relatif konsisten. Cocok untuk tren jangka menengah, tapi kurang baik untuk pergerakan ekstrem. Lalu pada window 36 (3 tahun ke belakang), menghasilkan nilai MAPE sebesar 10,13% yang artinya fokus prediksi lebih kepada tren panjang, tapi sering lagging berat atau bahkan salah arah di ujung periode. Meskipun bisa mengikuti tren besar, kesalahan di periode akhir bisa sangat tinggi. Error lebih tidak konsisten → MAPE cenderung lebih besar dibanding window 12 & 24.

V. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan, dapat disimpulkan telah diketahui hasil dari masing masing pengukuran window size. Dari penghitungan evaluasi dan prediksi model diperoleh hasil MAPE (Mean Absolute Percentage Error) sebesar 8,53% untuk window size 24, kemudian 6,58% untuk window size 12, dan 10,13% untuk window size 36. Dari hasil evaluasi metrik tersebut dapat diketahui jika ukuran window size 12 merupakan ukuran yang mendekati akurat untuk mengukur akurasi atau kinerja model karena memiliki persentase error paling kecil di antara window size 24 dan window size 36. Tujuan pelatihan selama beberapa periode (epoch) adalah supaya

model dapat melihat data pelatihan (train) berulang kali dan secara bertahap menyesuaikan parameter internalnya (bobot dan bias) untuk meminimalkan fungsi loss dan meningkatkan kemampuannya dalam membuat prediksi yang akurat. EarlyStopping callbacks membantu menghentikan pelatihan lebih awal jika performa model pada set validasi berhenti meningkat, sehingga mencegah overfitting.. Perbedaan Amplitudo jika garis oranye menangkap bentuk umum garis biru tetapi puncak dan lembahnya kurang jelas atau lebih ekstrem, model tersebut mungkin tidak sepenuhnya menangkap besarnya fluktuasi harga. Lonjakan dan Penurunan, apakah model dapat memprediksi kenaikan atau penurunan harga yang tiba-tiba. Window size juga memengaruhi akurasi prediksi karena mengatur seberapa banyak pola atau data masa lalu yang dipakai pada model. Jika window terlalu pendek, model kehilangan pola penting. Sedangkan jika window terlalu panjang, model bisa overfitting atau justru menangkap noise. Untuk perbaikan error / kinerja model, bisa ditambahkan data lain yang relevan (misalnya, faktor musiman, kebijakan pemerintah, harga komoditas lain), bisa dipertimbangkan untuk menambahkannya sebagai fitur input. Langkah yang bisa dilakukan untuk memperbaiki error yaitu bisa mencoba jumlah unit yang berbeda di layer LSTM dan Dense. Kemudian menyesuaikan tingkat dropout untuk mencegah overfitting. Dan bisa juga tambahkan layer LSTM atau Dense tambahan jika dirasa perlu untuk menangkap pola yang lebih kompleks.

REFERENSI

- [1] Cahyani, J., Mujahidin, S., & Fiqar, T. P. (2023). Implementasi Metode Long Short Term Memory (LSTM) untuk Memprediksi Harga Bahan Pokok Nasional. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, 11(2), 346. <https://doi.org/10.26418/justin.v11i2.57395>
- [2] Santoso, A., Purnamasari, A. I., & Ali, I. (n.d.). PREDIKSI HARGA BERAS MENGGUNAKAN METODE RECURRENT NEURAL NETWORK DAN LONG SHORT-TERM MEMORY.
- [3] Badieah, B., Gernowo, R., & Surarso, B. (2016). Metode Jaringan Syaraf Tiruan Untuk Prediksi Performa Mahasiswa Pada Pembelajaran Berbasis Problem Based Learning (PBL). *JURNAL SISTEM INFORMASI BISNIS*, 6(1), 46. <https://doi.org/10.21456/vol6iss1pp46-58>
- [4] Hidayah, T. (n.d.). Perbandingan Algoritma Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) untuk Prediksi Harga Emas Dunia.
- [5] Sianturi, T. B., Cholissodin, I., & Yulistira, N. (2023). Penerapan Algoritma Long Short-Term Memory (LSTM) berbasis Multi Fungsi Aktivasi Terbobot dalam Prediksi Harga Ethereum (Vol. 7, Issue 3). <http://j-ptiik.ub.ac.id>
- [6] Mukhlisin, Y., Imrona, M., & Mardiansyah, D. T. (n.d.). Prediksi Harga Beras Premium dengan Metode Algoritma K-Nearest Neighbor.
- [7] Setiadi, D. (n.d.). Analisis Prediksi Harga Beras Berbasis Kualitas Menggunakan Algoritma K-Nearest Neighbord.
- [8] Nanja, M. (2015). METODE K-NEAREST NEIGHBOR BERBASIS FORWARD SELECTION UNTUK PREDIKSI HARGA KOMODITI LADA. In *Jurnal Pseudocode* (Vol. 2, Issue 1). www.ejournal.unib.ac.id53
- [9] Suahati, A. F., Nurrahman, A. A., & Rukmana, O. (2022). Penggunaan Jaringan Syaraf Tiruan – Backpropagation dalam Memprediksi Jumlah Mahasiswa Baru. *Jurnal Media Teknik Dan Sistem Industri*, 6(1), 21. <https://doi.org/10.35194/jmtsi.v6i1.1589>
- [10] Model Penetapan Harga Beras di Pasar Tradisional Oleh, P., Fernando Nainggolan, M., Saragi, C. P., Sitohang, M., Agribisnis, P., Pertanian, F., & Santo Thomas, U. (n.d.). Model Penetapan Harga

Beras di Pasar Tradisional.
<http://ejournal.ust.ac.id/index.php/AGRIUST>

- [11] M. M'Saad, M. Bouslimani, and M. A. Latifi, "PID adaptive control of exothermic stirred tank reactors," in *Proc. 2nd IEEE Conference on Control Applications*, Sep. 1993, pp. 113–117.
- [12] R. Murray and B. Grunloh. (view Jan. 2013). FAQ: what is steady-state error? [Online]. Available: http://www.cds.caltech.edu/~murray/amwiki/index.php/FAQ:_What_is_steady_state_error%3F.