

**OPTIMALISASI PREDIKSI HARGA BERAS DI PROVINSI
SUMATERA UTARA MENGGUNAKAN MODEL *HYBRID*
SARIMA-LSTM**

SKRIPSI

DANIEL MANURUNG

201401044



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2025**

**OPTIMALISASI PREDIKSI HARGA BERAS DI PROVINSI SUMATERA
UTARA MENGGUNAKAN MODEL *HYBRID* SARIMA-LSTM**

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana
Ilmu Komputer

DANIEL MANURUNG

201401044



PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2025

PERSETUJUAN

Judul : OPTIMALISASI PREDIKSI HARGA BERAS DI PROVINSI SUMATERA UTARA MENGGUNAKAN MODEL HYBRID SARIMA-LSTM

Kategori : SKRIPSI

Nama : DANIEL MANURUNG

Nomor Induk Mahasiswa : 201401044

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

Telah diuji dan dinyatakan lulus di Medan, 9 Januari 2025

Komisi Pembimbing :

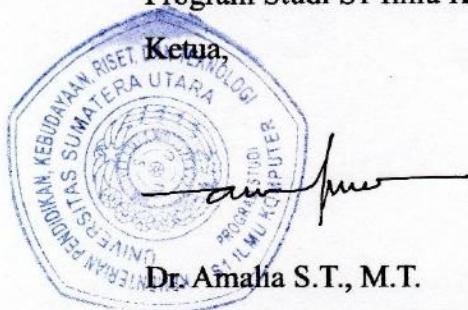
Dosen Pembimbing II

Dosen Pembimbing I

Pauzi Ibrahim Nainggolan S.Komp., M.Sc. Dr. T. Henny Febriana Harumy S.Kom., M.Kom
NIP. 198809142020011001 NIP. 198802192019032016

Diketahui/disetujui oleh

Program Studi S1 Ilmu Komputer



Dr. Amalia S.T., M.T.

NIP. 197812212014042001

PERNYATAAN**OPTIMALISASI PREDIKSI HARGA BERAS DI PROVINSI SUMATERA
UTARA MENGGUNAKAN MODEL *HYBRID SARIMA-LSTM*****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 9 Januari 2025



Daniel Manurung

NIM. 201401044

UCAPAN TERIMA KASIH

Puji dan syukur ke hadirat Tuhan YME atas berkat dan kasih karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir skripsi ini sebagai syarat memperoleh gelar sarjana Ilmu Komputer pada program studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih kepada para pihak yang telah memberikan bantuan dan bimbingan kepada penulis selama proses penyusunan tugas akhir skripsi ini, antara lain:

1. Bapak Prof. Dr. Muryanto Amin S.Sos., M.Si., selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
3. Ibu Dr. Amalia S.T., M.T., selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi, Univesitas Sumatera Utara.
4. Ibu Dr. T. Henny Febriana Harumy S.Kom., M.Kom, sebagai dosen pembimbing I dan Bapak Pauzi Ibrahim Nainggolan S.Komp., M.Sc. sebagai dosen pembimbing II yang telah memberikan bimbingan, saran, dan masukan kepada penulis selama proses penyusunan skripsi ini.
5. Ibu Dr. Ir. Elviawaty Muisa Zamzami S.T., M.T., M.M., IPU, selaku dosen pembimbing akademik yang telah memberikan bimbingan akademik kepada penulis selama menjalani perkuliahan.
6. Seluruh dosen Program Studi S-1 Ilmu Komputer yang telah memberikan pengetahuan dan bimbingan kepada penulis selama masa perkuliahan.
7. Seluruh staf dan pegawai di Fakultas Ilmu Komputer dan Teknologi Informasi, atas bantuan administrasi yang diberikan selama masa perkuliahan.
8. Kedua orang tua yang tercinta, yaitu Jono Tani Manurung dan Manipat Panjaitan yang memberikan dukungan moral dan materil, doa, motivasi, dan semangat kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.

9. Abang dan kakak kandung penulis, yaitu Yosua B. J. Manurung, Grace Meilisa Br. Manurung, dan Willy Rivaldo Manurung yang memberikan doa dan semangat kepada penulis selama proses penyusunan skripsi.
10. Sahabat – sahabat penulis, yaitu John Tri Putra Sihombing, Jonathan Lexi Febrian Sitohang, dan Rio Fransiskus Simanjuntak, yang telah memberikan dukungan dan semangat selama proses penyusunan skripsi ini.
11. Seluruh teman dan rekan KOM B 2020 dan Stambuk 2020 S-1 Ilmu Komputer yang telah membantu dan mendukung penulis dalam menyelesaikan tugas akhir skripsi ini.
12. Semua pihak lain yang telah membantu dan memberikan dukungan dalam proses penyusunan skripsi ini, yang tidak dapat disebutkan satu per satu oleh penulis.

Terima kasih atas segala bimbingan dan dukungan selama proses penyusunan skripsi ini. Semoga Tuhan Yang Maha Esa senantiasa memberkati kita semua. Akhir kata, semoga skripsi ini dapat bermanfaat bagi para pembaca.

Medan, 5 Januari 2025



Daniel Manurung
NIM. 201401044

OPTIMALISASI PREDIKSI HARGA BERAS DI PROVINSI SUMATERA UTARA MENGGUNAKAN MODEL HYBRID SARIMA-LSTM

ABSTRAK

Beras merupakan bahan makanan pokok masyarakat Indonesia, sehingga kestabilan harga dan ketersediannya harus terjaga. Jika harga beras mengalami kenaikan, tentunya akan memberikan dampak yang buruk karena akan berpengaruh pada peningkatan inflasi. Untuk itu prediksi harga beras penting untuk dilakukan untuk mendukung pengambilan keputusan oleh pembuat kebijakan, penjual, dan konsumen. Penelitian ini bertujuan untuk mengembangkan sebuah *website* yang dapat melakukan prediksi harga beras di provinsi Sumatera Utara. Untuk menghasilkan prediksi yang lebih optimal, model gabungan SARIMA dan LSTM akan dibangun untuk memanfaatkan kelebihan dari kedua model. Pengujian dilakukan dengan membandingkan performa model individu SARIMA dan LSTM dengan model *Hybrid* SARIMA-LSTM. Hasil penelitian menunjukkan model SARIMA cenderung kurang akurat dalam menangkap pola non-linear pada data aktual. Di sisi lain, LSTM lebih baik dalam menangkap fluktuasi harga, namun masih memiliki *error* yang cukup tinggi pada beberapa prediksi. Model *Hybrid* memberikan hasil yang paling optimal dengan nilai MAE 24,168 dan MAPE 0,158% untuk beras premium, serta MAE 21,680 dan MAPE 0,155% untuk beras medium.

Kata Kunci : Prediksi, Beras, SARIMA, LSTM, *Hybrid* SARIMA-LSTM

OPTIMIZATION OF RICE PRICE PREDICTION IN NORTH SUMATRA PROVINCE USING THE SARIMA-LSTM HYBRID MODEL

ABSTRACT

Rice is a staple food for the people of Indonesia, making it crucial to maintain price stability and availability. A rise in rice prices can have adverse effects as it contributes to increased inflation. Therefore, rice price forecasting is essential to support decision-making by policymakers, sellers, and consumers. This research aims to develop a website that can predict rice prices in North Sumatra province. To achieve optimal predictions, a hybrid model combining SARIMA and LSTM will be built to utilize the advantages of both models. Tests were conducted by comparing the performance of individual SARIMA and LSTM models with the Hybrid SARIMA-LSTM model. The results show that the SARIMA model tends to be less accurate in capturing the non-linear patterns of actual data. Meanwhile, the LSTM model performs better in capturing price fluctuations but still exhibits significant errors in some predictions. The Hybrid model provides the most optimal results, with an MAE 24,168 and MAPE of 0,158% for premium rice, and an MAE 21,680 and MAPE of 0,155% for medium rice.

Keywords: Prediction, Rice, SARIMA, LSTM, Hybrid SARIMA-LSTM

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN.....	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian	3
1.6.1 Studi Pustaka.....	3
1.6.2 Analisa dan perencanaan.....	4
1.6.3 Implementasi.....	4
1.6.4 Pengujian.....	4
1.6.5 Dokumentasi	4
1.7 Sistematika Penulisan.....	4
BAB 2 LANDASAN TEORI.....	6

2.1	Peramalan Deret Waktu	6
2.2	<i>Autoregressive Integrated Moving Average (ARIMA)</i>	7
2.2.1	SARIMA	7
2.2.2	Stasioneritas	8
2.2.3	Uji Augmented Dickey-Fuller (ADF)	9
2.2.4	Autocorrelation function dan partial autocorrelation function	10
2.2.5	Auto-ARIMA	10
2.3	<i>LSTM (Long Short-Term Memory)</i>	11
2.3.1	Multi-step ahead prediction	14
2.3.2	Normalisasi dan denormalisasi	14
2.4	<i>Hybrid SARIMA-LSTM</i>	15
2.5	Beras	15
2.6	Evaluasi Performa Model	16
2.7	Streamlit	16
BAB 3	PERANCANGAN SISTEM	18
3.1	Kebutuhan Sistem	18
3.1.1	Hardware	18
3.1.2	Software	18
3.2	Perancangan Sistem	19
3.2.1	Pengumpulan Dataset	20
3.2.2	Pra-pemrosesan data	20
3.2.3	Implementasi model SARIMA	21
3.2.4	Implementasi model LSTM	22
3.2.5	Implementasi model Hybrid SARIMA-LSTM	23
3.2.6	Evaluasi performa model	24
3.2.7	Implementasi sistem	24
BAB 4	IMPLEMENTASI DAN PENGUJIAN	26

4.1	Pengumpulan Dataset	26
4.2	Visualisasi Data	27
4.3	Pra-pemrosesan Data	28
4.4	Implementasi Model SARIMA	29
4.4.1	Seasonal decompose	29
4.4.2	Uji stasioneritas.....	30
4.4.3	Identifikasi parameter model SARIMA.....	31
4.4.4	Pemilihan model terbaik	33
4.4.5	Prediksi dan Evaluasi.....	35
4.5	Implementasi Model LSTM	39
4.5.1	Pra-pemrosesan data untuk model LSTM.....	39
4.5.2	Pelatihan model LSTM	41
4.5.3	Pengujian dan evaluasi.....	42
4.6	Implementasi Model <i>Hybrid</i> SARIMA-LSTM	47
4.6.1	Pengambilan residual dari model SARIMA	47
4.6.2	Pelatihan model LSTM	47
4.6.3	Pengujian dan evaluasi.....	49
4.7	Evaluasi	53
4.8	Implementasi sistem	54
BAB 5	PENUTUP	58
5.1	Kesimpulan.....	58
5.2	Saran.....	59
	DAFTAR PUSTAKA	60
	LAMPIRAN	63

DAFTAR GAMBAR

Gambar 2.3.1 Arsitektur LSTM (Kang, et al., 2020)	11
Gambar 3.2.1 Alur pengembangan sistem	19
Gambar 4.1.1 Panel harga pangan	26
Gambar 4.2.1 Grafik pergerakan harga beras premium	27
Gambar 4.2.2 Grafik pergerakan harga beras medium	28
Gambar 4.3.1 Dataset harga beras	28
Gambar 4.3.2 Informasi dataset	29
Gambar 4.8.1 Tampilan awal <i>website</i> prediksi harga beras	54
Gambar 4.8.2 Visualisasi harga beras dengan <i>line chart</i>	55
Gambar 4.8.3 Proses prediksi harga beras pada <i>website</i>	55
Gambar 4.8.4 Grafik hasil prediksi <i>out of sample</i> harga BP 14 hari kedepan	56
Gambar 4.8.5 Grafik hasil prediksi <i>out of sample</i> harga BM 14 hari kedepan.....	56
Gambar 3.2.1.1 <i>Website</i> panel harga pangan	20
Gambar 3.2.3.1 Alur pengembangan model SARIMA.....	21
Gambar 3.2.4.1 Alur pengembangan model LSTM.....	22
Gambar 3.2.5.1 Alur pengembangan model <i>Hybrid</i> SARIMA-LSTM.....	23
Gambar 4.4.1.1 Grafik seasonality beras premium.....	29
Gambar 4.4.1.2 Grafik seasonality beras medium	29
Gambar 4.4.3.1 Grafik ACF dan PACF (beras premium).....	32
Gambar 4.4.3.2 Grafik ACF dan PACF (beras medium)	32
Gambar 4.4.4.1 Visualisasi pembagian data model SARIMA (BP)	33
Gambar 4.4.4.2 Visualisasi pembagian data model SARIMA (BM)	33
Gambar 4.4.4.3 Hasil <i>auto-arima</i> beras premium	34
Gambar 4.4.4.4 Hasil <i>auto-arima</i> beras medium.....	34
Gambar 4.4.5.1 Grafik perbandingan aktual vs prediksi harga BP (SARIMA)	35
Gambar 4.4.5.2 Grafik perbandingan aktual vs prediksi harga BM (SARIMA)	35
Gambar 4.5.1.1 Visualisasi pembagian data LSTM.....	41
Gambar 4.5.3.1 Grafik perbandingan aktual vs prediksi harga BP (LSTM)	43
Gambar 4.5.3.2 Grafik perbandingan aktual vs prediksi harga BM (LSTM)	43

Gambar 4.6.3.1 Visualisasi pembagian data untuk LSTM pada model <i>Hybrid</i>	50
Gambar 4.6.3.2 Grafik perbandingan aktual vs prediksi harga BP (<i>Hybrid</i>).....	50
Gambar 4.6.3.3 Grafik perbandingan aktual vs prediksi harga BM (<i>Hybrid</i>)	51

DAFTAR TABEL

Tabel 2.6.1 Signifikansi MAPE (Chang, Wang, & Liu, 2007)	16
Tabel 4.5.1 Nilai variabel untuk contoh perhitungan LSTM	44
Tabel 4.7.1 Perbandingan nilai <i>error</i> seluruh model.....	53
Tabel 2.2.4.1 Proses analisis grafik ACF dan PACF (As'ad, Wibowo, & Sophia, 2017)	10
Tabel 4.4.2.1 Pengujian ADF data beras	30
Tabel 4.4.2.2 Pengujian ADF setelah transformasi dan differencing pertama.....	31
Tabel 4.4.5.1 Hasil prediksi harga beras dengan model SARIMA	36
Tabel 4.4.5.2 Metrik evaluasi prediksi harga beras.....	39
Tabel 4.5.1.1 Data setelah normalisasi.....	39
Tabel 4.5.1.2 Data input untuk beras premium	40
Tabel 4.5.1.3 Data output untuk beras premium	40
Tabel 4.5.2.1 Daftar pengujian parameter LSTM	41
Tabel 4.5.2.2 Kombinasi parameter terbaik untuk prediksi harga beras	42
Tabel 4.5.3.1 Prediksi 14 hari kedepan dengan model LSTM	44
Tabel 4.5.3.2 Metrik evaluasi model LSTM	46
Tabel 4.6.1.1 <i>Residual</i> model SARIMA	47
Tabel 4.6.2.1 Kombinasi parameter terbaik untuk <i>training set</i> 80%	48
Tabel 4.6.2.2 Kombinasi parameter terbaik untuk <i>training set</i> 70%	48
Tabel 4.6.2.3 Kombinasi parameter terbaik untuk <i>training set</i> 60%	48
Tabel 4.6.3.1 Pengujian model <i>Hybrid</i> SARIMA-LSTM.....	49
Tabel 4.6.3.2 Hasil prediksi harga beras dengan model <i>Hybrid</i>	51
Tabel 4.6.3.3 Metrik evaluasi model <i>Hybrid</i>	52

DAFTAR LAMPIRAN

Lampiran 1 Plot perbandingan ketiga model <i>hybrid</i> (Beras Premium)	63
Lampiran 2 Plot perbandingan ketiga model <i>hybrid</i> (Beras Medium)	63
Lampiran 3 Perbandingan hasil prediksi seluruh model.....	64
Lampiran 4 Hasil Prediksi 1 Juni 2024 – 14 Juni 2024	64
Lampiran 5 Hasil Prediksi 1 Juni 2024 – 14 Juni 2024	65
Lampiran 6 <i>Source Code</i> Pengembangan <i>Website</i>	65

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Beras merupakan komoditas pangan utama yang menjadi makanan pokok sehari-hari masyarakat Indonesia. Hal tersebut membuat Indonesia menjadi konsumen beras terbesar keempat di dunia dengan angka 35,3 juta metrik ton pada 2022/2023 (Muhamad, 2023). Sebagai kebutuhan pokok masyarakat Indonesia, harga beras dan ketersediaan beras haruslah terjaga.

Menurut (Ohyver & Pudjihastuti, 2018) harga beras di Indonesia cenderung dipengaruhi oleh siklus panen musiman. Biasanya saat musim panen tiba, harga beras cenderung turun karena kuantitas beras yang meningkat. Sebaliknya saat di luar musim panen, harga cenderung naik, hal ini dikarenakan permintaan beras yang besar tidak sebanding dengan ketersediaan beras. Kenaikan harga beras sendiri tentunya akan memberikan dampak yang buruk karena dapat berpengaruh terhadap peningkatan inflasi (Azwina & Syahbudi, 2023).

Menurut informasi yang diperolah dari Badan Pusat Statistik (BPS), tingkat inflasi bulanan beras pada Februari 2024 mencapai 5,32%. Angka ini meningkat dibanding dengan inflasi bulanan beras pada Januari 2024 sebesar 0,64%. Hal ini lantaran berbagai jenis beras di seluruh rantai distribusi di Indonesia mengalami kenaikan harga rerata nasional sebesar 5,28 % di bulan Februari 2024 (Widi, 2024). Jika inflasi terjadi maka banyak dampak buruk yang akan terjadi seperti meningkatnya angka kemiskinan, terganggunya stabilitas keamanan dan politik. Oleh karena itu, prediksi harga beras penting untuk dilakukan. Hal ini akan sangat berguna untuk para pembuat kebijakan, penjual, dan konsumen. Dengan prediksi harga yang akurat akan memberikan dampak positif dalam pengelolaan pasokan beras dan pengendalian inflasi (Barrett, 1997).

Beberapa penelitian tentang prediksi harga beras telah dilakukan sebelumnya. Penelitian yang dilakukan oleh (Lastinawati, Mulyana, Zahri, & Sriati, 2019) membahas pengembangan model ARIMA untuk meramalkan harga beras di Kabupaten Ogan Komering Ilir, Provinsi Sumatera Utara. Hasil dari penelitian ini menjelaskan bahwa harga eceran beras premium dan medium di Kabupaten OKI akan terus meningkat hingga Desember 2020. Selanjutnya penelitian oleh (Fadil Indra Sanjaya, 2020) membahas penggunaan model LSTM dalam memprediksi rerata harga beras di tingkat grosir di Indonesia. Hasil penelitian ini menunjukkan bahwa model LSTM dapat memprediksi harga beras dengan tingkat ketepatan yang tinggi dengan nilai RMSE adalah 0,43. Penelitian lain dilakukan oleh (Visakha & Wustqa, 2023) membahas peramalan harga beras di Indonesia menggunakan metode *hybrid* ARIMA dan *Neural Network* untuk mendapatkan model peramalan terbaik. Hasilnya menunjukkan bahwa model ini memiliki akurasi tinggi, dengan nilai MAPE sebesar 0,9778%.

Dalam rangka untuk memprediksi harga beras di Provinsi Sumatera Utara, model *hybrid* dari penggabungan dua algoritma yaitu SARIMA dan LSTM akan digunakan. Baik SARIMA dan LSTM memiliki kelebihannya masing-masing. SARIMA mampu menangani pola linear pada data deret waktu dengan baik (Chen & Wang, 2007). Di sisi lain LSTM mampu mengidentifikasi pola non-linear dengan baik (Siami-Namini, Tavakoli, & Namin, 2018). Kedua algoritma ini akan digabungkan untuk mendapatkan model prediksi yang paling optimal dimana dapat memperoses pola linear dan non-linear dalam data. Model yang telah dibangun nantinya akan diintegrasikan kedalam website agar hasil peramalan dapat dilihat oleh semua kalangan. Pemodelan yang dilakukan menggunakan data historis harga beras sebelumnya. Data yang akan digunakan merupakan data harian. Data dikumpulkan dalam urutan waktu atau disebut juga dengan data deret waktu (*Time Series Data*).

1.2 Rumusan Masalah

Beras merupakan makanan pokok sehari-hari masyarakat Indonesia. Hal ini menjadikan Indonesia menjadi konsumen beras terbesar keempat di dunia. Kenaikan harga beras sendiri tentunya akan memberikan dampak yang buruk seperti potensi terjadinya inflasi. Prediksi harga beras penting untuk dilakukan untuk dapat membantu para pembuat kebijakan membuat perencanaan kedepannya. Pada penelitian ini akan dilakukan

penggabungan dua algoritma peramalan yaitu SARIMA dan LSTM untuk melakukan prediksi harga beras di Provinsi Sumatera Utara.

1.3 Batasan Masalah

Berikut adalah beberapa batasan masalah dalam penelitian ini.

1. Data harga beras diperoleh dari sistem informasi harga pangan strategis Badan Pangan Nasional.
2. Fokus penelitian ini pada prediksi harga beras di Provinsi Sumatera Utara
3. Beras yang akan diprediksi terdiri dari dua jenis, yaitu beras premium dan beras medium
4. Model dirancang dengan menggunakan bahasa Python
5. Program dirancang berbasis website.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan aplikasi berbasis *web* yang dapat melakukan prediksi harga beras premium dan harga beras medium secara harian selama 14 hari kedepan di Provinsi Sumatera Utara.

1.5 Manfaat Penelitian

Berikut beberapa manfaat dari penelitian ini.

1. Memberikan informasi prediksi harga beras untuk petani dan produsen di Provinsi Sumatera Utara, sehingga mereka dapat mengoptimalkan produksi dan persediaan dengan lebih efisien
2. Prediksi harga beras yang lebih akurat dapat digunakan pemerintah untuk membuat kebijakan untuk pengendalian harga beras seperti mengatur kebijakan fiskal dan perencanaan anggaran.

1.6 Metodologi Penelitian

1.6.1 *Studi Pustaka*

Penelitian dimulai dengan mengumpulkan referensi dari berbagai sumber yang kredibel dan melakukan tinjauan literatur melalui artikel ilmiah , jurnal, *e-book*, makalah, serta *website* yang relevan dengan topik penelitian.

1.6.2 *Analisa dan perencanaan*

Penulis melakukan analisis terhadap berbagai kebutuhan penelitian berdasarkan ruang lingkup yang telah ditetapkan. Hasil analisis ini kemudian disusun menjadi diagram alir (*flowchart*) untuk memvisualisasikan proses penelitian secara keseluruhan.

1.6.3 *Implementasi*

Setelah dilakukan analisa dan perancangan, model SARIMA, LSTM, dan *Hybrid* SARIMA-LSTM akan dibuat, mengikuti diagram alir yang telah disusun sebelumnya.

1.6.4 *Pengujian*

Model yang telah dibuat kemudian dilakukan pengujian untuk memprediksi harga beras. Pengujian mencakup evaluasi kinerja model menggunakan metrik evaluasi.

1.6.5 *Dokumentasi*

Penelitian yang telah dilaksanakan, dilakukan pencatatan menyeluruh dari setiap langkah yang dilakukan selama penelitian.

1.7 Sistematika Penulisan

BAB 1 PENDAHULUAN

Bab ini menguraikan latar belakang penelitian, perumusan masalah, batasan masalah, tujuan, manfaat penelitian, metodologi penelitian, serta sistematika penulisan skripsi.

BAB 2 LANDASAN TEORI

Bab ini memaparkan teori yang relevan dengan penelitian ini, khususnya mengenai prediksi harga beras menggunakan model SARIMA, LSTM dan *Hybrid* SARIMA-LSTM. Landasan teori berfungsi sebagai referensi yang mendukung perancangan sistem predikksi harga beras berbasis *website* yang akan dikembangkan.

BAB 3 PERANCANGAN SISTEM

Bab ini menguraikan proses perancangan sistem yang akan dibuat dalam penelitian ini. Proses perancangan dimulai dengan mengidentifikasi kebutuhan sistem, merancang alur pengembangan sistem, dan penjelasan masing-masing alur.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab ini menguraikan implementasi model berdasarkan rancangan yang telah disusun sebelumnya, serta proses pengujian untuk mengevaluasi apakah model sudah cukup baik untuk diintegrasikan ke dalam *website*

BAB 5 PENUTUP

Bab terakhir ini berisi kesimpulan dari penelitian yang telah dilakukan serta saran dari penulis untuk penelitian yang berkaitan dengan topik yang sama di kemudian hari.

BAB 2

LANDASAN TEORI

2.1 Peramalan Deret Waktu

Data deret waktu adalah deretan data hasil pengamatan dalam rentang waktu tertentu yang memiliki interval yang konsisten, seperti jam, hari, minggu, bulan, tahun. Konsep dasar dari deret waktu adalah pengamatan sekarang (Z_t) dipengaruhi oleh pengamatan sebelumnya (Z_{t-k}) (Wulandari & Gernowo, 2019). Analisis deret waktu adalah prosedur yang dilakukan untuk memprediksi struktur probabilistik di masa mendatang dalam rangka pengambilan keputusan (Nurfadilah, C, & Kasse, 2018).

Peramalan deret waktu adalah proses untuk menghasilkan suatu prediksi informasi di masa depan berdasarkan data historis. Hasil dari peramalan deret waktu dapat berguna bagi pihak yang membutuhkan data tersebut untuk membuat suatu keputusan (Wiranda & Sadikin, 2019). Peramalan sering digunakan di berbagai bidang, contohnya di sektor pertanian untuk memprediksi harga pangan. Bidang bisnis dan ekonomi untuk memprediksi harga saham. Dalam meteorologi untuk memprediksi suhu harian dan curah hujan.

Untuk menerapkan metode peramalan deret waktu, penting untuk memahami pola dari data tersebut. Menurut (Makridakis, Wheelwright, & McGee, 1999, as cited in Deviana, Nusyirwan, Azis, & Ferdias, 2021), terdapat empat jenis pola dalam data deret waktu.

1. Pola Horizontal

Pola ini terjadi ketika data menunjukkan kestabilan tanpa tren naik atau turun yang signifikan.

2. Pola Musiman

Pola ini muncul ketika fluktuasi data dipengaruhi oleh faktor yang berulang secara teratur berdasarkan waktu, seperti musim dalam setahun, dan bulan.

3. Pola Trend

Pola ini menggambarkan arah pergerakan data dalam jangka waktu yang panjang, baik itu naik atau turun.

4. Pola Siklis

Muncul ketika data dipengaruhi oleh fluktuasi ekonomi dalam periode waktu yang panjang

2.2 Autoregressive Integrated Moving Average (ARIMA)

ARIMA adalah algoritma yang dikembangkan oleh dua peneliti, yaitu George Box dan Gwilyn Jenkins. Metode ini sendiri memiliki nama lain yaitu metode Box-Jenkins. Dimana nama ini diambil dari nama kedua peneliti tersebut. ARIMA adalah metode yang populer untuk peramalan data deret waktu *univariat* (Hariadi & Sulantari, 2021).

Model ARIMA merupakan kombinasi dari tiga komponen, yaitu model *autoregressive* (AR), model *moving average* (MA) dan pembedaan (I). Jika nonstasionaritas diatasi dengan menambahkan proses pembedaan (I) pada model ARMA, maka diperoleh bentuk umum ARIMA(p,d,q). Di mana p adalah orde AR, d adalah jumlah diferensiasi yang dilakukan untuk mencapai keadaan stasioner, dan q adalah orde MA. Persamaan model ARIMA dapat dituliskan sebagai berikut.

$$\phi_p(B)(1 - B)^d Y_t = c + \theta_q(B)e_t \quad (1)$$

Dimana :

c	= Konstanta
e_t	= galat pada periode ke-t
$(1 - B)^d$	= proses pembedaan orde ke-d
$\phi_p(B)$	= $(1 - \phi B - \phi B^2 - \dots - \phi B^p)$ yaitu operator langkah mundur untuk AR
$\theta_q(B)$	= $(1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)$ yaitu operator langkah mundur untuk MA

2.2.1 SARIMA

Seasonal Autoregressive Integrated Moving Average (SARIMA) adalah pengembangan dari model ARIMA dengan penambahan parameter *seasonality* (Ridla, Azise, & Rahman, 2023). Jika data deret waktu menunjukkan pola musiman (*seasonality*), maka

model SARIMA adalah pilihan yang lebih tepat untuk digunakan. Model SARIMA dapat dinyatakan dalam notasi sebagai berikut.

$$\text{ARIMA } (p, d, q)(P, D, Q)^s$$

Dimana :

(p, d, q) : Komponen non-musiman

(P, D, Q) : Komponen musiman

s : Periode musiman

Bentuk umum model SARIMA dapat ditulis seperti dibawah :

$$\Phi_P(B^s)\phi_p(B)(1 - B^s)^D(1 - B)^d y_t = \Theta_Q(B^s)\theta_q(B)\varepsilon_t \quad (2)$$

Dimana :

ε_t : Nilai galat pada waktu ke-t

y_t : Nilai pengamatan pada waktu ke-t

$(1 - B)^d$: Operator *differencing* non-musiman

$(1 - B^s)^D$: Operator *differencing* musiman

$\phi_p(B)$: Operator *autoregressive*

$\Phi_P(B^s)$: Operator *autoregressive* musiman

$\theta_q(B)$: Operator *moving average*

$\Theta_Q(B^s)$: Operator *moving average* musiman

Pemilihan parameter terbaik untuk model SARIMA adalah dengan membandingkan nilai *Akaike Information Criterion* (AIC). AIC digunakan untuk menentukan apakah model sudah bisa memprediksi dengan baik dan juga cukup sederhana. Model dengan nilai AIC yang paling kecil menunjukkan model tersebut memiliki kinerja prediktif yang lebih baik (Zhao, Li, & Qu, 2022).

2.2.2 Stasioneritas

Stasioner adalah suatu kondisi dimana data bersifat stabil atau tidak mengalami perubahan yang signifikan selama periode waktu tertentu (Deviana, et al., 2021). Stasioneritas menjadi syarat dalam pemodelan ARIMA dan SARIMA. Apabila data tidak menunjukkan kondisi stasioner, maka diperlukan transformasi atau diferensiasi

untuk membuatnya stasioner. Data deret waktu disebut stasioner bila memiliki *mean* dan *variance* yang konstan seiring waktu .

Stasioner dalam *mean* berarti nilai *mean* data tetap konstan seiring waktu. Bila data tidak stasioner dalam *mean*, maka perlu ditambahkan proses diferensiasi. Diferensiasi (∇^d) adalah sebuah Teknik untuk mengubah data menjadi stasioner dengan menghitung selisih antara nilai pada pengamatan tertentu dan pada pengamatan sebelumnya. Berikut adalah persamaan *differencing* (Ardesfira, et al., 2022).

$$\nabla^d = X_i - X_{i-d} = (1 - B)X_i \quad (3)$$

Di mana X_t merupakan nilai pengamatan pada waktu ke- i dan B adalah operator *backshift* dengan persamaan.

$$B^k X_i = X_{i-k} \quad (4)$$

Stasioner dalam varians berarti nilai varians data tetap konstan seiring waktu (Sitorus, Wahyuningsih, & Hayati, 2017). Bila data deret waktu tidak menunjukkan kondisi stasioner pada *variance*, maka diperlukan transformasi salah satunya dengan metode Box-Cox (Wei, 2006, as cited in Deviana, et al., 2021). Transformasi Box-Cox merupakan metode yang menerapkan pangkat λ pada variabel dependen Y. Dengan demikian, bentuk transformasi dengan metode ini adalah Y^λ , di mana λ merupakan parameter yang harus diestimasi. Metode yang dapat digunakan untuk mengestimasi λ adalah metode Kemungkinan Maksimum. Berikut ini persamaan transformasi Box-Cox (Yati, Devianto, & Asdi, 2013).

$$W_i(\lambda) = \begin{cases} \left(\frac{Y_i^\lambda - 1}{\lambda} \right), \lambda \neq 0 \\ \ln(Y_i), \lambda = 0 \end{cases}, i = 1, 2, \dots, n \quad (5)$$

2.2.3 Uji Augmented Dickey-Fuller (ADF)

Uji ADF merupakan metode uji statistik yang disebut juga uji akar unit. Tujuan dari pengujian ini adalah untuk menentukan apakah data terdapat akar unit atau tidak, yang membantu dalam menilai apakah data tersebut stasioner. Pengujian ADF dilakukan dengan membandingkan *p-value* dengan level signifikansi 5% (0,05) atau dengan membandingkan nilai ADF *test statistic* dengan seluruh *critical values*. Bila *p-value* \leq

0,05, maka H_0 ditolak dan data dianggap stasioner. Sebaliknya, bila $p\text{-value} > 0,05$, H_0 diterima, dan data dianggap tidak stasioner. Jika statistik uji ADF > nilai kritis, maka H_0 diterima dan data dianggap tidak stasioner. Sedangkan jika statistik uji ADF < nilai kritis, maka H_0 ditolak dan dapat dikatakan data deret waktu stasioner (Sirisha, Belavagi, & Attigeri, 2022).

2.2.4 Autocorrelation function dan partial autocorrelation function

Autocorrelation Function (ACF) adalah fungsi yang mengukur korelasi antara data pada waktu ke- i dengan data pada waktu sebelumnya (Deviana, et al., 2021). Sedangkan, *Partial Autocorrelation Function* (PACF) adalah fungsi yang menghitung korelasi data Y_t dan Y_{t+k} ($k = \text{lag}$) namun sebelumnya menghilangkan data lain diantara kedua data tersebut, yaitu Y_{t+1} sampai dengan Y_{t+k-1} (Rosadi, 2011, as cited in Deviana, et al., 2021).

Dalam proses pemilihan parameter untuk ARIMA atau SARIMA, grafik ACF dan PACF digunakan untuk menentukan orde AR, MA, ARMA. Tabel 2.2.4.1 menunjukkan proses analisis untuk memilih nilai p dan q (P dan Q).

Tabel 2.2.4.1 Proses analisis grafik ACF dan PACF (As'ad, Wibowo, & Sophia, 2017)

Orde	ACF	PACF
AR(p)	Menurun menuju nol secara eksponensial	Tiba-tiba turun menuju nol setelah lag p
MA(q)	Tiba-tiba turun menuju nol setelah lag q	Menurun menuju nol secara eksponensial
ARMA(p,q)	Menurun menuju nol	Menurun menuju nol

2.2.5 Auto-ARIMA

Auto-ARIMA adalah sebuah fungsi dari *package* pmdarima yang digunakan untuk mengidentifikasi parameter paling optimal untuk model ARIMA atau SARIMA. Cara kerjanya sama dengan *grid search*, dimana setiap kombinasi yang mungkin dari nilai parameter yang diberikan pengguna akan diuji.

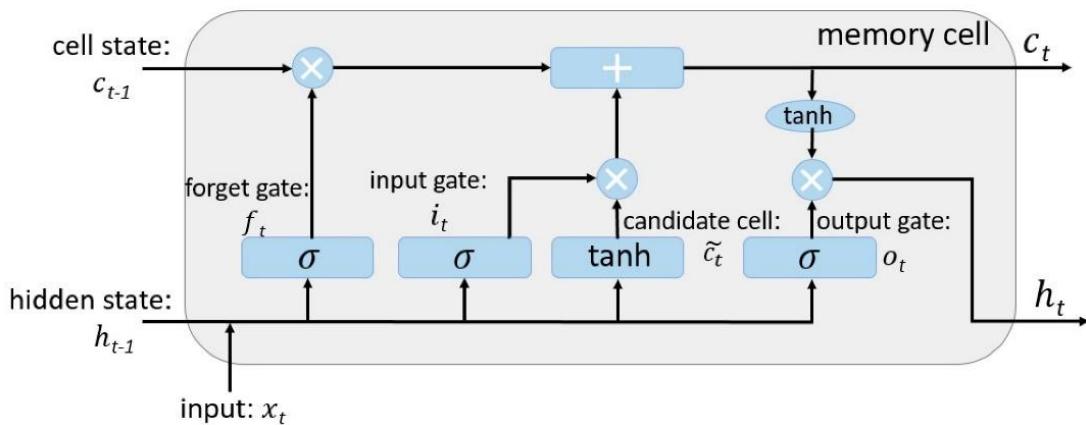
Sebelum memulai pemilihan model, *Auto-ARIMA* akan menggunakan tes stasioner untuk menentukan orde *differencing d* maupun *D*. Selanjutnya Fungsi ini akan mencoba berbagai kombinasi dari parameter p dan q (juga P dan Q untuk parameter

musiman), kemudian setiap model akan dibandingkan untuk memilih model terbaik berdasarkan kriteria yang dipilih (AIC atau BIC). Pada penelitian ini akan digunakan kriteria AIC.

2.3 LSTM (*Long Short-Term Memory*)

LSTM pertama kali diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997. Arsitektur LSTM merupakan perkembangan dari arsitektur RNN, yang mana tujuan pengembangannya untuk mengatasi masalah *vanishing gradient* yang ditemukan pada RNN konvensional. *Vanishing gradient* adalah sebuah kondisi dimana nilai gradien selalu mengecil hingga mencapai layer terakhir, sehingga bobot tidak mengalami perubahan dan menyebabkan model tidak dapat mencapai hasil yang lebih baik atau tidak konvergen (Rowan, Muflikhah, & Cholissodin, 2022). Untuk mengatasi permasalahan *vanishing gradient* ini LSTM memiliki *cell state* yang berfungsi untuk menyimpan atau melupakan informasi. *Cell state* memiliki struktur yang disebut *cell gates*, yang terbagi menjadi 4 komponen, yaitu *input gate*, *forget gate*, *memory-cell state gate*, dan *output gate* (Poomka, Pongsena, Kerdprasop, & Kerdprasop, 2019).

Arsitektur dari LSTM ditampilkan pada gambar 2.3.1.



Gambar 2.3.1 Arsitektur LSTM (Kang, et al., 2020)

1. *Forget gate*

Forget gate berfungsi untuk memilih informasi apa yang harus dihapus dari *memory* dengan menggunakan layer sigmoid (Kang, et al., 2020). Berikut ini adalah persamaan dari *forget gate* (Pierre, Akim, Semenyo, & Babiga, 2023).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (6)$$

Dimana :

W_f : Matriks bobot untuk f_t

b_f : Bias untuk f_t

x_t : Input pada waktu t

h_{t-1} : *Hidden state* dari waktu sebelumnya

σ : Fungsi sigmoid

. Hasil dari persamaan diatas adalah sebuah nilai antara 0 dan 1. Untuk nilai 1 merepresentasikan informasi dari *cell state* (C_{t-1}) sebelumnya akan dipertahankan. Untuk nilai 0 merepresentasikan informasi dari *cell state* sebelumnya akan dibuang.

2. *Input Gate*

Input gate dan *candidate cell* akan menentukan informasi baru apa yang akan disimpan di *cell state* (Kang, et al., 2020). Berikut adalah persamaan *input gate* (i_t) dan *candidate cell* (\tilde{C}_t) (Pierre, Akim, Semenyo, & Babiga, 2023).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (7)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (8)$$

Dimana :

W_i : Matriks bobot untuk i_t

b_i : Bias untuk i_t

x_t : Input pada waktu t

h_{t-1} : *Hidden state* dari waktu sebelumnya

W_c : Vektor bobot untuk \tilde{C}_t

b_c : Bias untuk \tilde{C}_t

\tanh : Fungsi aktivasi tanh

3. Cell State

Cell state(C_t) baru kemudian dibentuk agar dapat digunakan jaringan untuk menghilangkan informasi yang tidak dibutuhkan dan memasukkan informasi baru (Kang, et al., 2020). Berikut ini adalah persamaan dari C_t (Kang, et al., 2020).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (9)$$

Dimana :

f_t : *Forget gate*

C_{t-1} : *Cell state*

i_t : *Input gate*

\tilde{C}_t : *Candidate cell*

4. Output Gate

Output gate (o_t) berfungsi untuk menentukan informasi dari memori yang akan digunakan sebagai *output* (Kang, et al., 2020) (Pierre, Akim, Semenyo, & Babiga, 2023). Berikut adalah persamaan *output gate* (Kang, et al., 2020).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (10)$$

$$h_t = o_t * \tanh(C_t) \quad (11)$$

W_o : Matriks bobot untuk o_t

b_o : Bias untuk o_t

x_t : Input pada waktu t

h_{t-1} : *Hidden state* dari waktu sebelumnya

σ : Fungsi sigmoid

h_t : *Hidden state baru*

C_t : *Cell state*

\tanh : Fungsi tanh

2.3.1 *Multi-step ahead prediction*

Dalam penelitian, prediksi harga beras akan dilakukan selama 14 hari kedepan. Untuk melakukan prediksi beberapa langkah tersebut perlu diterapkan *multi-step ahead prediction*. Untuk melakukan prediksi beberapa langkah kedepan maka akan digunakan suatu strategi yaitu *Direct Strategy*.

Pada *Direct Strategy* sebanyak n model akan dilatih secara independen untuk memprediksi masing-masing n langkah kedepan. Misalnya, jika diperlukan prediksi untuk 2 langkah kedepan, maka dua model akan dilatih, model pertama akan memprediksi langkah pertama dan model kedua akan memprediksi langkah kedua. Perlu diperhatikan bahwa prediksi pada titik kedua tidak dipengaruhi oleh hasil estimasi dari titik pertama. Kelebihan dari strategi ini adalah nilai *error* tidak akan terakumulasi karena setiap model tidak akan menggunakan hasil prediksi model lain sebagai input (Masum, Liu, & Chiverton, 2018).

2.3.2 *Normalisasi dan denormalisasi*

Data deret waktu seharusnya tidak dijadikan input secara langsung kepada model. Untuk mendapatkan hasil pelatihan yang lebih baik, data harus dilakukan normalisasi (Hu, Zhang, Ding, & Zhang, 2022). Pada penelitian ini normalisasi dilakukan dengan teknik *min-max scaling*. Berikut adalah rumus dari *min-max scaling*.

$$x' = \frac{x_i - \min_x}{\max_x - \min_x} \quad (12)$$

Dimana x' adalah hasil normalisasi, x_i adalah data asli, \min_x adalah data terkecil, dan \max_x adalah data terbesar diantara seluruh data. Data yang telah dinormalisasi kemudian dibagi menjadi *input* dan *output* dengan menggunakan teknik *sliding window*. Misalnya jika data bulan Januari ingin dijadikan *output*, maka data berdimensi- n dibuat dengan menggunakan data dari n bulan sebelumnya sebagai data *input*.

Output dari LSTM bukanlah hasil akhir peramalan. Nilai asli *output* bisa didapat dengan melakukan *reverse normalization* atau Denormalisasi (Hu, Zhang, Ding, & Zhang, 2022). Denormalisasi adalah proses mengembalikan nilai yang telah di normalisasi kembali ke skala aslinya. Denormalisasi data dapat dinyatakan dengan persamaan dibawah ini.

$$x = x'(max_x - min_x) + min_x \quad (13)$$

2.4 Hybrid SARIMA-LSTM

Model SARIMA telah sangat berhasil dalam penelitian dan dalam pengaplikasiannya di bidang industri. Namun model ini memiliki keterbatasan utama, dimana bentuk modelnya yang diasumsikan linear. Dengan kata lain, model ini menganggap adanya struktur korelasi linear di antara nilai-nilai deret waktu, sehingga SARIMA tidak mampu menangkap pola non-linear apa pun (Chen & Wang, 2007). Di sisi lain metode *deep learning* seperti model LSTM memiliki kemampuan mengidentifikasi struktur dan pola non-linear pada data deret waktu (Siami-Namini, Tavakoli, & Namin, 2018). Untuk mendapatkan hasil prediksi yang paling optimal, maka penggabungan dua model ini dilakukan untuk mendapatkan model yang dapat menangani baik pola linear dan non-linear dalam data.

Data deret waktu dapat dibagi menjadi dua bagian, yaitu struktur linear dan struktur non-linear. Proses pelatihan model *Hybrid* SARIMA-LSTM dibagi menjadi dua bagian, yaitu proses pelatihan model SARIMA yang mana akan memproses komponen linear dan proses pelatihan model LSTM yang akan memproses *residual* dari model SARIMA atau komponen non-linear. Komponen *residual* merupakan selisih antara data aktual dan hasil prediksi. Setelah kedua model selesai dilatih, tahap selanjutnya adalah peramalan. SARIMA akan melakukan peramalan sebanyak n hari dan LSTM akan melakukan peramalan *residual* sebanyak n hari. Hasil dari kedua model kemudian digabungkan untuk mendapatkan hasil akhir (Yu, Feng, Li, & Cheng, 2022).

2.5 Beras

Beras adalah salah satu komoditas pangan utama yang menjadi makanan pokok sehari-hari orang Indonesia. Di tingkat konsumen atau eceran terdapat dua jenis beras yang dijual, yaitu beras premium dan beras medium. Keduanya dibedakan berdasarkan kualitasnya. Kualitas kedua jenis beras tersebut dibedakan kedalam beberapa parameter, yaitu sosoh, kadar air, beras kepala, beras patah, serta jumlah butir beras lainnya yang mencakup butir menir, merah, kuning/rusak, kapur, butir gabah dan benda asing lainnya (Anwar, 2021).

Dari parameter tersebut beras premium dan beras medium memiliki tingkat yang sama pada derajat sosoh dan kadar air. Namun untuk parameter lainnya beras premium

memiliki nilai yang lebih baik, sehingga menjadikannya berkualitas lebih tinggi dibandingkan beras medium. Oleh karena itu, harga beras premium juga lebih mahal dibandingkan dengan beras medium (Anwar, 2021).

2.6 Evaluasi Performa Model

Hasil prediksi biasanya memiliki nilai *error*, sehingga evaluasi kinerja model diperlukan untuk mengetahui seberapa baik akurasi prediksi dari model. Dalam penelitian ini, penulis menggunakan dua metrik *error* prediksi, yaitu MAE dan MAPE.

Mean Absolute Error (MAE) digunakan untuk mengukur rata-rata selisih absolut antara nilai aktual dan nilai yang di prediksi. Sementara itu, MAPE (*Mean Absolute Percentage Error*) digunakan untuk mengukur rata-rata selisih persentase absolut antara data aktual dan hasil prediksi. Semakin kecil nilai kedua metrik ini, semakin akurat prediksi yang dilakukan oleh model. Berikut ini persamaan dari MAE dan MAPE (Ardesfira, et al., 2022).

$$MAE = \frac{1}{n} \sum_{t=1}^n |X_t - \hat{X}_t| \quad (14)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{X_i - \hat{X}_i}{X_i} \right| * 100 \quad (15)$$

Di mana n adalah jumlah data, X_i adalah nilai aktual, dan \hat{X}_i adalah hasil prediksi. Adapun kriteria signifikansi MAPE tercantum pada tabel 2.6.1.

Tabel 2.6.1 Signifikansi MAPE (Chang, Wang, & Liu, 2007)

MAPE	Signifikansi
< 10%	Sangat Baik
10 – 20%	Baik
20 – 50%	Cukup
> 50%	Buruk

2.7 Streamlit

Streamlit merupakan *framework* Python sumber terbuka yang dirancang untuk membuat *user interface* interaktif untuk aplikasi *data science*. Streamlit memungkinkan

pengembangan aplikasi *web* dengan Python secara responsif dan dinamis tanpa memerlukan banyak kode (Maulid, 2023).

Streamlit memungkinkan seorang *data scientist* untuk dengan cepat membangun aplikasi interaktif yang memvisualisasi data dan menjalankan model *machine learning*. Hal ini dikarenakan streamlit mudah di integrasikan dengan berbagai *framework* Python lain untuk *data science*, seperti Pandas untuk analisis data, Plotly untuk visualisasi data, Keras untuk pemodelan, dan banyak lagi (Maulid, 2023).

Fitur lainnya yang sangat berguna adalah adanya *Streamlit Community Cloud*. Dengan ini pengguna dapat melakukan *deploy* untuk aplikasi streamlit yang telah dibuat secara gratis. Fitur *deploy* ini terhubung langsung dengan repositori GitHub sehingga setiap perubahan yang dilakukan pada kode di repositori, akan langsung diterapkan pada *website* sehingga memudahkan pengguna karena tidak perlu melakukan *deploy* berulang kali (Snowflake Inc., n.d.).

BAB 3

PERANCANGAN SISTEM

3.1 Kebutuhan Sistem

Untuk melakukan pengembangan model prediksi dan aplikasi Prediksi Harga Beras di Provinsi Sumatera Utara berbasis *website*, peneliti membutuhkan beberapa komponen perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk menunjang pengembangan sistem ini. Berikut ini adalah daftar komponen yang dibutuhkan.

3.1.1 Hardware

Berikut ini daftar *hardware* yang dibutuhkan pada penelitian ini :

1. *Processor* AMD Ryzen 5 3500U.
2. *Memory* RAM 8 GB.
3. *Graphics* AMD Radeon Vega 8.
4. *Storage* SSD NVME.

3.1.2 Software

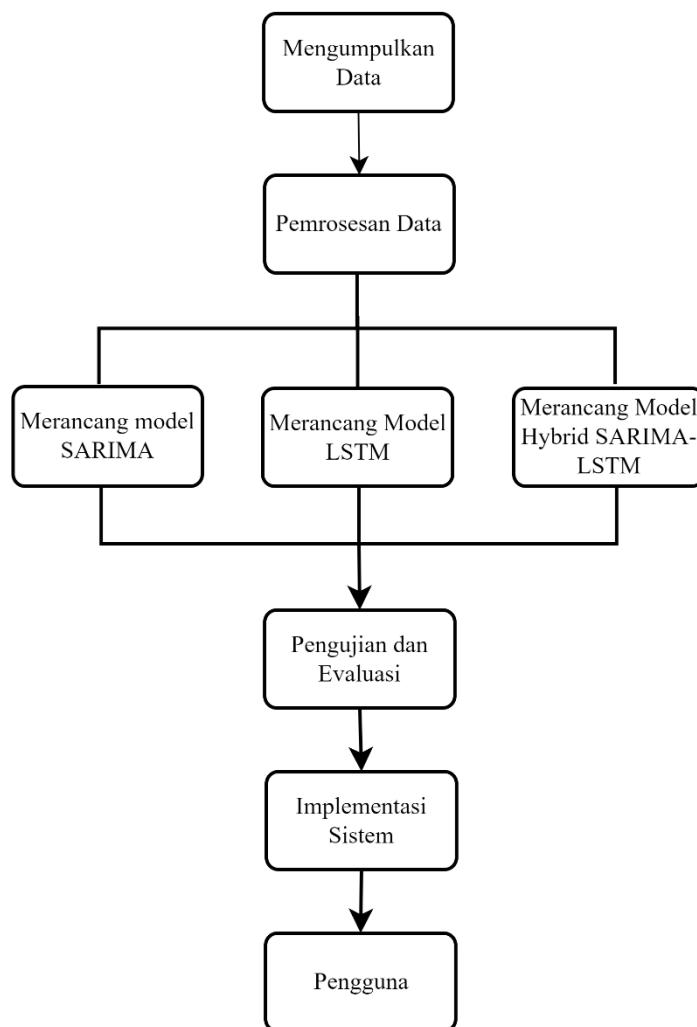
Berikut ini daftar *software* yang dibutuhkan pada penelitian ini :

1. Windows 11 sebagai operasi sistem.
2. Google Collab sebagai platform pengembangan model *machine learning*.
3. Visual Studio Code sebagai IDE untuk pengembangan *website*.
4. Python 3 sebagai bahasa pemrograman untuk pengembangan model dan *website*.
5. *Library* dan *framework* Python (Pandas, Keras, Streamlit) sebagai tools untuk pengembangan model dan *website*.
6. GitHub sebagai platform untuk menyimpan *source code*.
7. Streamlit Community Cloud sebagai platform *hosting* yang digunakan untuk *men-deploy website*.

3.2 Perancangan Sistem

Perancangan sistem mengikuti langkah-langkah pengembangan sistem yang ditunjukkan pada gambar 3.2.1. Proses dimulai dari pengumpulan data historis harga beras premium dan medium. Data kemudian diproses untuk memastikan data *input* bersih dan siap digunakan untuk pelatihan model. Selanjutnya akan dilakukan pelatihan model SARIMA, LSTM, dan *Hybrid* SARIMA-LSTM.

Setelah pelatihan, model dievaluasi dengan menggunakan metrik evaluasi untuk mengukur performa dan menentukan apakah model *Hybrid* SARIMA-LSTM memberikan hasil yang lebih optimal. Tahap terakhir dari proses ini adalah pengembangan antarmuka pengguna berupa *website* dengan mengintegrasikan model *Hybrid* SARIMA-LSTM.



Gambar 3.2.1 Alur pengembangan sistem

3.2.1 Pengumpulan Dataset

Data yang dipakai dalam penelitian ini diperoleh dari laman *web* Badan Pangan Nasional (BPN). Sumber ini dipilih karena BPN merupakan lembaga pemerintah Republik Indonesia, sehingga data yang disediakan dapat dipastikan akurat. Data yang diambil adalah harga beras premium dan beras medium dengan periode waktu dimulai dari Juli 2022 sampai dengan Mei 2024.

Gambar 3.2.1.1 Website panel harga pangan

Gambar 3.2.1.1 merupakan antarmuka panel harga pangan yang disediakan oleh *website* resmi Badan Pangan Nasional. Untuk mendapatkan data harga pangan, pengguna dapat mengisi kolom Provinsi, Kabupaten, dan Tanggal untuk menyesuaikan tempat dan rentang tanggal untuk data yang ingin dilihat. Selanjutnya untuk mengunduh data pengguna dapat menekan tombol “Download Tabel”.

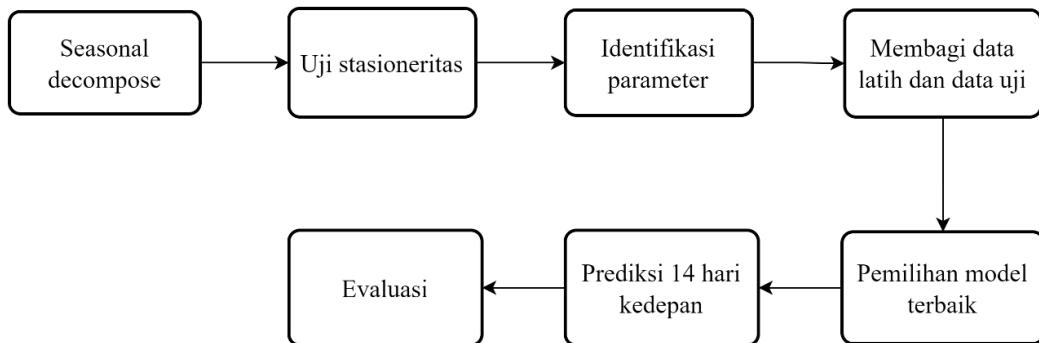
3.2.2 Pra-pemrosesan data

Dataset yang dikumpulkan kemudian menjalani tahap pra-pemrosesan untuk mempersiapkan data agar siap digunakan dalam pelatihan model. Tahap ini bertujuan untuk meningkatkan kualitas data dan memastikan bahwa data yang digunakan dalam model sudah bersih. Pra-pemrosesan data dapat melibatkan pembersihan data, seperti penanganan data hilang, penanganan outlier. Selain pembersihan data, tahap ini juga dapat melibatkan transformasi data dan *differencing* untuk menghilangkan tren dan memastikan data menjadi stasioner dalam pemodelan SARIMA. Untuk pemodelan LSTM tahap pra-pemrosesan data meliputi normalisasi untuk mengubah skala data.

Teknik *sliding window* digunakan untuk membagi data deret waktu menjadi data input dan output untuk pelatihan model LSTM.

3.2.3 Implementasi model SARIMA

Setelah dilakukan tahap pra-pemrosesan data, selanjutnya adalah tahap pemodelan. Untuk model pertama yang akan dibuat adalah model SARIMA.



Gambar 3.2.3.1 Alur pengembangan model SARIMA

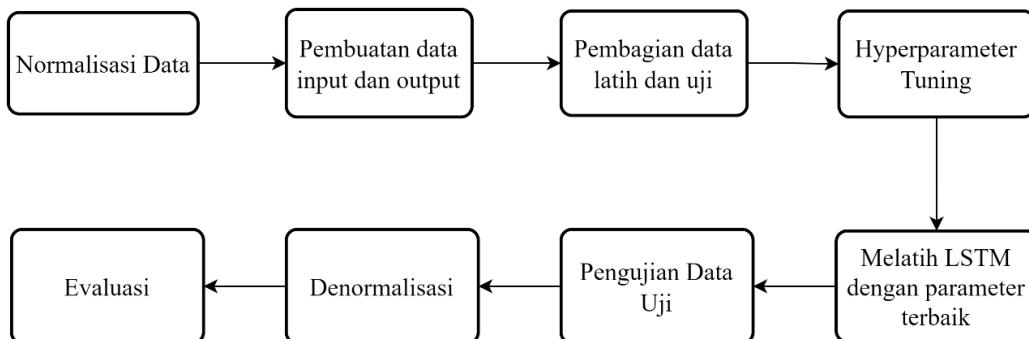
Berikut ini adalah tahapan pengembangan model SARIMA yang ditunjukkan pada gambar 3.2.3.1 diatas :

1. Menggunakan *seasonal decompose* untuk memecah data deret waktu menjadi tiga komponen, yaitu *trend*, *seasonal*, dan *residual*. Tujuannya adalah untuk mengidentifikasi apakah data menunjukkan pola musiman atau tidak. Jika ada indikasi musiman, model SARIMA akan menjadi pilihan yang lebih tepat.
2. Berikutnya adalah menguji stasioneritas data. Bila data tidak stasioner, maka dilakukan transformasi menggunakan metode Box-Cox dan *differencing*.
3. Mengidentifikasi parameter model untuk komponen non-musiman (*p*, *d*, *q*) dan komponen musiman (*P*, *D*, *Q*). Proses identifikasi parameter dilakukan melalui analisis plot ACF dan PACF.
4. Sebanyak 14 data terakhir dalam dataset akan dipisahkan sebagai data uji dan tidak digunakan dalam pelatihan model. Data uji akan digunakan untuk mengukur kinerja atau performa model dalam memprediksi nilai di masa depan.
5. Melakukan pemilihan model terbaik dengan mencoba semua kombinasi parameter yang mungkin untuk memperoleh model dengan kinerja prediksi yang terbaik. Pemilihan model dilakukan dengan membandingkan metrik AIC. Model dengan nilai AIC yang lebih rendah akan dianggap memiliki performa prediksi yang lebih baik.

6. Menguji model dengan memprediksi 14 hari kedepan untuk melihat performa model dalam memprediksi data baru. Hasil prediksi kemudian dievaluasi dengan menggunakan metrik MAE dan MAPE.

3.2.4 Implementasi model LSTM

Model kedua yang akan dibuat adalah model LSTM. Untuk memprediksi 14 hari kedepan, pengembangan model LSTM akan menggunakan *Direct Strategy*, dimana satu model akan dilatih untuk menghasilkan keluaran sebanyak 14 data.



Gambar 3.2.4.1 Alur pengembangan model LSTM

Berikut ini adalah tahapan pengembangan model LSTM yang ditunjukkan pada gambar 3.2.4.1 diatas:

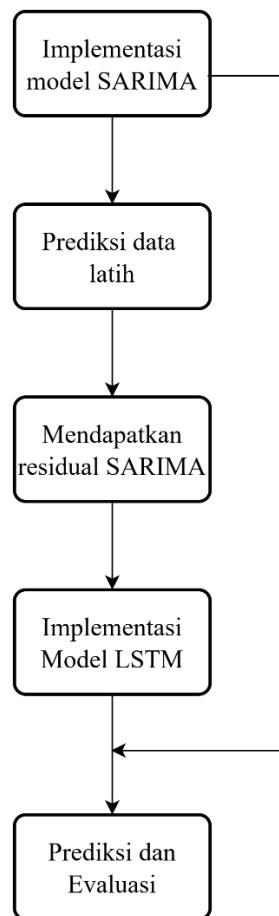
1. Data yang telah bersih selanjutnya di normalisasi dengan menggunakan teknik *min-max scaling* dengan rentang 0,0 hingga 1,0
2. Setelah itu dengan menerapkan fungsi *sliding window* data dibagi menjadi data input dan output. Untuk *window size* yang akan digunakan adalah 5 dan untuk jumlah kolom output adalah 14 dimana setiap kolom digunakan untuk kolom output 14 model.
3. Langkah berikutnya adalah melakukan pembagian data latih dan uji. Data uji terdiri dari 14 baris terakhir data yang telah diterapkan *sliding window* akan dijadikan data uji untuk menguji performa model memprediksi data baru. Setelah itu dilakukan *reshaping* pada data latih untuk mengubah dimensinya menjadi 3 dimensi agar dapat digunakan untuk pelatihan LSTM.
4. Untuk mendapatkan *hyperparameter* terbaik, maka akan dilakukan *hyperparameter tuning*. Tahap ini akan menguji setiap kombinasi parameter LSTM untuk mendapatkan model dengan performa prediksi yang paling akurat. Sebanyak

20% data terakhir pada data latih akan dipakai sebagai data validasi. Data ini berguna untuk menilai performa model dalam memprediksi data baru.

5. Parameter terbaik dari proses tuning kemudian digunakan untuk melatih 14 model LSTM. Sama seperti proses *hyperparameter tuning* sebanyak 20% data training akan dijadikan data validasi.
6. Selanjutnya dilakukan pengujian prediksi untuk 14 hari kedepan. Hasil prediksi kemudian dilakukan *Denormalisasi* untuk mengembalikan nilainya ke skala asli. Hasil prediksi kemudian dievaluasi dengan menggunakan metrik MAE dan MAPE.

3.2.5 Implementasi model Hybrid SARIMA-LSTM

Setelah parameter terbaik untuk SARIMA dan LSTM didapatkan, selanjutnya adalah melakukan pemodelan *Hybrid* SARIMA-LSTM.



Gambar 3.2.5.1 Alur pengembangan model *Hybrid* SARIMA-LSTM

Berikut ini adalah tahapan pengembangan model *Hybrid* SARIMA-LSTM yang ditunjukkan pada gambar 3.2.5.1 diatas :

1. Model SARIMA dilatih dengan parameter terbaik yang telah didapat sebelumnya, kemudian dilakukan prediksi pada data latih.
2. Nilai *residual* dari prediksi SARIMA kemudian dihitung. Nilai *residual* didapat dengan mengambil selisih antara data aktual dengan hasil prediksi. Nilai *residual* kemudian dijadikan data latih untuk model LSTM.
3. Selanjutnya adalah implementasi model LSTM. Pada pemodelan *Hybrid* pembagian data latih dan data validasi akan dilakukan dalam 3 rasio pembagian, 80:20, 70:30, dan 60:40. Sebelum masuk ke proses pelatihan, dilakukan *hyperparameter tuning* untuk mendapatkan kombinasi parameter terbaik. Setelah didapatkan parameter terbaik, model LSTM kemudian dilatih.
4. Selanjutnya kedua model, SARIMA dan LSTM digunakan untuk melakukan prediksi. Hasil prediksi LSTM kemudian dilakukan denormalisasi untuk mengembalikannya ke skala aslinya. Prediksi kedua model kemudian digabungkan untuk mendapatkan hasil akhir. Performa Model *Hybrid* kemudian dievaluasi dengan menggunakan metrik MAE dan MAPE.

3.2.6 *Evaluasi performa model*

Setelah pelatihan model selesai, selanjutnya adalah mengevaluasi kinerja model untuk menilai keakuratannya dalam melakukan prediksi selama 14 hari kedepan. Tujuan evaluasi ini adalah untuk menentukan apakah model *Hybrid* SARIMA-LSTM memberikan performa yang paling optimal dibandingkan model lainnya. Evaluasi dilakukan menggunakan dua metrik, yaitu MAE dan MAPE. Model dengan nilai *error* terkecil dianggap memiliki performa terbaik. Hasil evaluasi ini akan menentukan apakah model *Hybrid* SARIMA-LSTM sudah layak untuk di integrasikan ke dalam website prediksi harga beras.

3.2.7 *Implementasi sistem*

Tahap terakhir adalah pengembangan antarmuka pengguna berupa *website*. Pembuatan *website* menggunakan bahasa pemrograman Python dan *framework* Streamlit. Streamlit dipilih karena *framework* ini memudahkan pengembangan *website* yang interaktif dan cepat. Streamlit juga memungkinkan integrasi langsung dengan berbagai *framework* Python yang juga digunakan dalam pengembangan model, seperti scikit-learn, pandas, keras, tensorflow, dan lain sebagainya. Dengan adanya fitur ini, akan memudahkan

untuk mengintegrasikan model yang telah dibuat dengan kecil kemungkinan akan adanya *error*.

Adapun fitur yang ada pada *website* prediksi harga beras ini, yaitu panel harga beras dengan pemilihan jenis beras dan rentang tanggal secara harian. Data harga beras ditampilkan dengan tabel dan dengan visualisasi *line chart*. Selain itu terdapat fitur prediksi harga beras yang akan melakukan prediksi selama 14 hari kedepan sesuai dengan jenis beras yang dipilih. Hasil prediksi akan ditampilkan dalam bentuk tabel dan *line chart*.

Setelah *website* selesai dibuat, selanjutnya akan dilakukan proses *deploying* menggunakan platform GitHub dan Streamlit Community Cloud. Tujuannya agar *website* dapat digunakan oleh khalayak umum. Untuk melakukan *deploying*, *source code* akan dimasukkan kedalam repositori GitHub. Selanjutnya akun GitHub akan ditautkan ke akun Streamlit agar repositori dapat diakses dan dilakukan *deploying*.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Pengumpulan Dataset

Dataset yang digunakan untuk pengembangan model prediksi diambil dari laman resmi Badan Pangan Nasional. Terdapat dua data yang diambil, yaitu harga beras premium dan medium. Data diambil dalam frekuensi harian dengan rentang tanggal dari 15 Juli 2022 sampai dengan 31 Mei 2024.

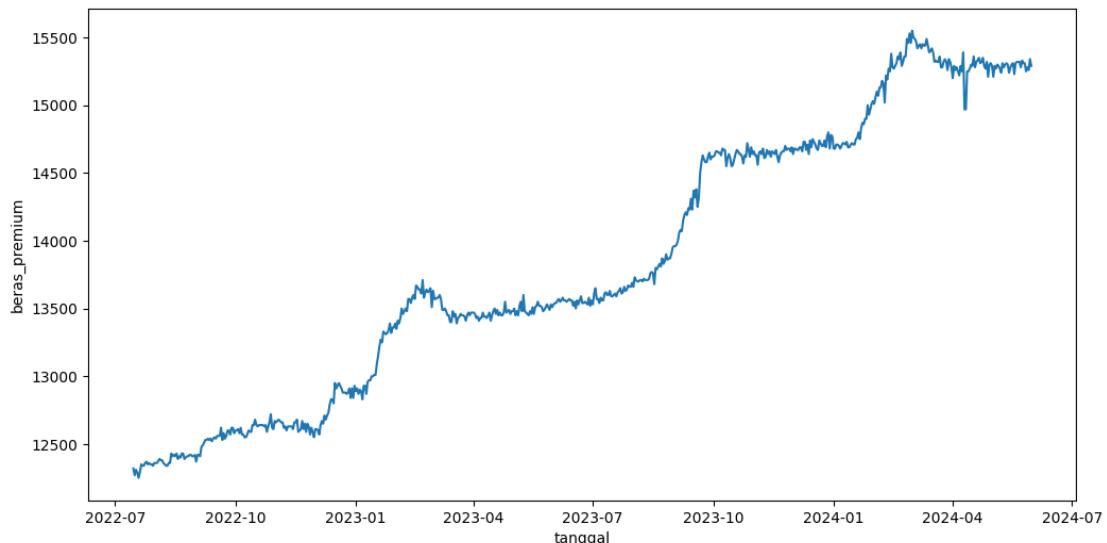
Komoditas (Rp)	15/07/2022	16/07/2022	17/07/2022	18/07/2022	19/07/2022	20/07/2022
Beras Premium	12.320	12.270	12.310	12.290	12.250	12.250
Beras Medium	11.020	10.950	11.060	11.020	11.020	11.020
Kedelai Biji Kering (Impor)	13.660	13.650	13.490	13.550	13.560	13.560
Bawang Merah	55.830	55.140	54.940	54.600	54.790	54.790
Bawang Putih Bonggol	26.000	25.990	25.850	25.860	25.660	25.660
Cabai Merah Keriting	85.570	85.160	84.710	86.690	87.030	87.030
Cabai Rawit Merah	89.440	89.630	93.810	83.380	77.070	77.070
Daging Sapi Murni	135.640	136.190	135.010	135.240	134.580	134.580
Daging Ayam Ras	31.750	31.650	31.660	31.300	31.530	31.530
Telur Ayam Ras	26.610	26.540	26.720	26.430	26.380	26.380
Gula Konsumsi	14.720	14.750	14.760	14.750	14.760	14.760
Minyak Goreng Kemasan Sederhana	21.270	20.860	20.740	21.020	20.870	20.870

Gambar 4.1.1 Panel harga pangan

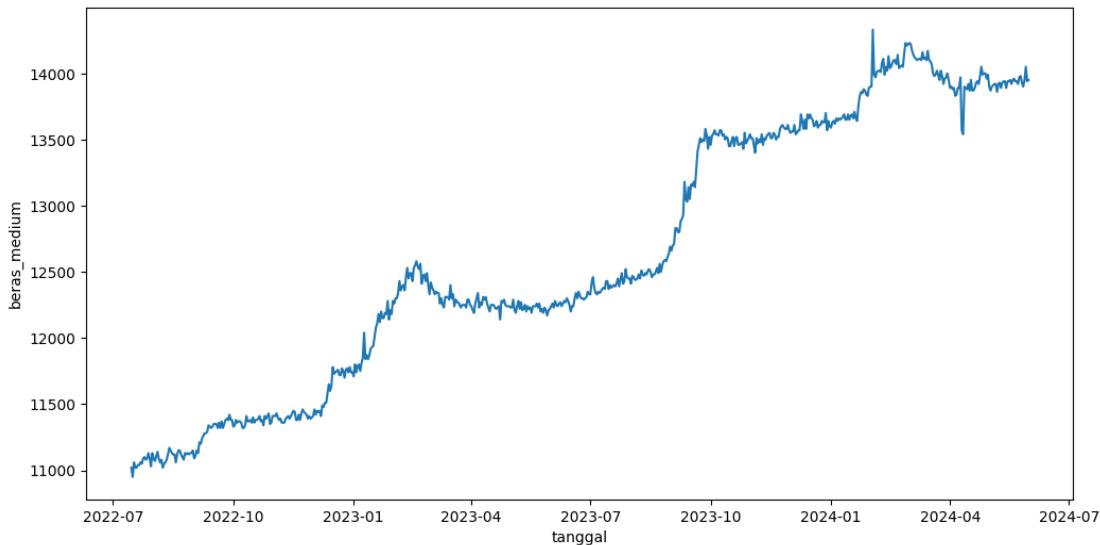
Pada gambar 4.1.1, provinsi yang dipilih adalah Sumatera Utara. Kabupaten yang dipilih adalah Semua Kabupaten/Kota. Tanggal yang dipilih dimulai dari 15-07-2022 sampai 31-05-2024. Setelah itu data di unduh dan hasil unduhan adalah dokumen dalam format .xlxs.

4.2 Visualisasi Data

Visualisasi data deret waktu berguna untuk memberikan wawasan mengenai pola yang ada pada data sebagai dasar untuk analisis lanjutan. Merujuk pada gambar 4.2.1 dan 4.2.2, harga beras premium dan beras medium menunjukkan tren kenaikan yang konsisten sepanjang periode pengamatan. Pada beras premium dan beras medium, harga cenderung mengalami kenaikan tajam di awal tahun. Hal ini tampak pada awal tahun 2023 dan 2024 yang mengalami kenaikan yang cukup tajam dan mulai menurun dari bulan maret.



Gambar 4.2.1 Grafik pergerakan harga beras premium



Gambar 4.2.2 Grafik pergerakan harga beras medium

4.3 Pra-pemrosesan Data

Pada tabel 4.1.1 terlihat *dataset* belum rapi dan masih banyak data harga pangan selain beras. Untuk itu perlu dilakukan *data cleaning* untuk menghilangkan data yang tidak dibutuhkan untuk pemodelan. Harga pangan lain akan dihapus dan struktur tabel akan diubah, jenis beras akan dijadikan kolom dan tanggal akan dijadikan baris.

	beras_premium	beras_medium
tanggal		
2022-07-15	12320	11020
2022-07-16	12270	10950
2022-07-17	12310	11060
2022-07-18	12290	11020
2022-07-19	12250	11020

Gambar 4.3.1 Dataset harga beras

Dapat dilihat pada gambar 4.3.1 *dataset* sudah lebih rapi dan data yang tidak diperlukan sudah hilang. Selanjutnya adalah melakukan pengecekan data hilang. Berdasarkan informasi pada gambar 4.3.2, dataset memiliki data yang hilang sebanyak 1 baris, baik pada beras premium dan beras medium. Untuk mengatasinya, *missing value* tersebut akan diisi dengan data lain dengan menggunakan teknik *linear interpolation*.

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 687 entries, 2022-07-15 to 2024-05-31
Freq: D
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   beras_premium    686 non-null   float64 
 1   beras_medium     686 non-null   float64 
dtypes: float64(2)
memory usage: 16.1 KB

```

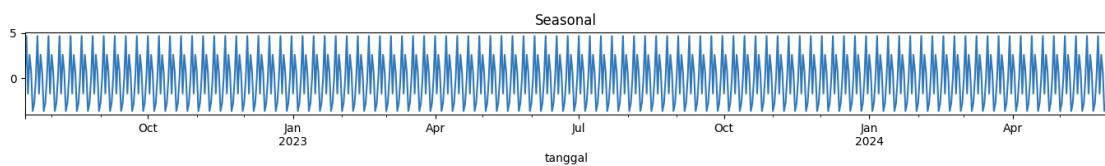
Gambar 4.3.2 Informasi dataset

4.4 Implementasi Model SARIMA

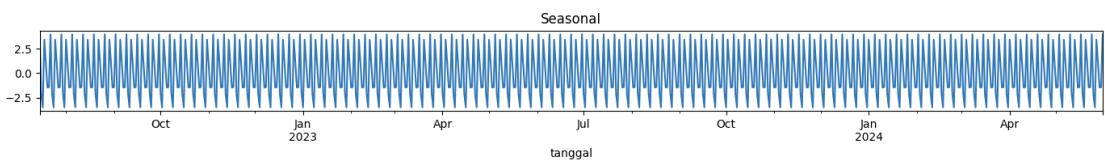
Pada tahap ini dilakukan pengembangan model SARIMA untuk memprediksi harga beras premium dan beras medium . Berikut ini tahapan implementasi model SARIMA.

4.4.1 Seasonal decompose

Tahap pertama yang dilakukan adalah membagi komponen pada data deret waktu menjadi 3 komponen, yaitu *trend*, *seasonal*, dan *residual* menggunakan metode *seasonal decompose* yang disediakan pustaka stasmodels. Untuk parameter period di isi 7 untuk melihat apakah ada musiman mingguan. Berikut ini adalah grafik *seasonality* untuk kedua jenis beras.



Gambar 4.4.1.1 Grafik seasonality beras premium



Gambar 4.4.1.2 Grafik seasonality beras medium

Dari gambar 4.4.1.1 dan 4.4.1.2, terlihat bahwa data harga dari kedua jenis beras memiliki komponen *seasonality* mingguan. Oleh sebab itu, model yang paling sesuai untuk kedua data tersebut adalah SARIMA.

4.4.2 *Uji stasioneritas*

Data yang akan digunakan untuk pelatihan model SARIMA haruslah stasioner. Berdasarkan gambar 4.2.1 dan 4.2.2, baik beras premium dan beras medium memiliki tren kenaikan yang konsisten sepanjang waku pengamatan, maka disimpulkan bahwa kedua data tidak stasioner.

Tabel 4.4.2.1 Pengujian ADF data beras

Jenis Beras		t-Statistic		Prob.*
Beras Premium	ADF test statistic		-0.5458576	0.8827528
	Critical values		1% -3.4400174	
	5% -2.8658060			
	10% -2.5690417			
Beras Medium	ADF test statistic		-0.9216781	0.7807497
	Critical values		1% -3.4401185	
	5% -2.8658505			
	10% -2.5690655			

Berdasarkan uji ADF pada tabel 4.4.2.1 dapat disimpulkan data beras premium dan beras medium tidak stasioner. Nilai ADF statistics pada kedua data tersebut lebih besar dari *critical value* (1%, 5%, 10%), maka H_0 diterima. Untuk itu perlu dilakukan transformasi Box-Cox dan diferensiasi.

Transformasi Box-Cox dilakukan dengan memanfaatkan fungsi stats.boxcox() yang disediakan pustaka Scipy. Hasil dari fungsi ini adalah data yang telah di transformasi dan nilai lambda yang berguna untuk membangkitkan nilai prediksi menjadi nilai asli. Tahap selanjutnya adalah melakukan *differencing* pada data yang telah di transformasi. Setelah dilakukan transformasi dan *differencing* pada data, selanjutnya adalah melakukan pengujian ADF untuk memastikan bahwa data telah stasioner. Pengujian ADF dapat dilihat pada tabel berikut.

Tabel 4.4.2.2 Pengujian ADF setelah transformasi dan differencing pertama

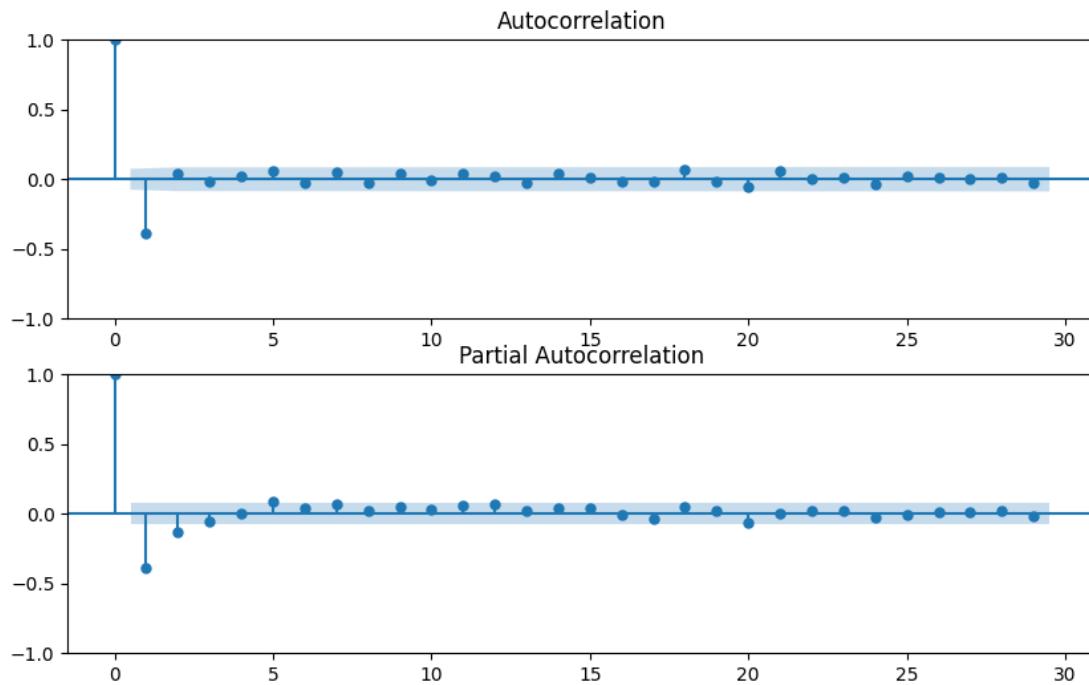
Jenis Beras		t-Statistic	Prob.*
Beras Premium	ADF test statistic	-8.9783508	7.425e-15
	Critical values	1% 5% 10%	-3.4400174 -2.8658060 -2.5690417
	ADF test statistic	-4.8816028	3.7866-05
	Critical values	1% 5% 10%	-3.4401185 -2.8658505 -2.5690655

Berdasarkan tabel 4.4.2.2 data beras premium setelah dilakukan transformasi dan *differencing* pertama menunjukkan nilai *p-value* adalah 7.425e-15 yang mana lebih rendah dari level signifikansi (0,05). Nilai ADF *test statistic* yang dihasilkan juga lebih rendah dari semua nilai *critical values*. Maka dapat disimpulkan data harga beras premium telah stasioner.

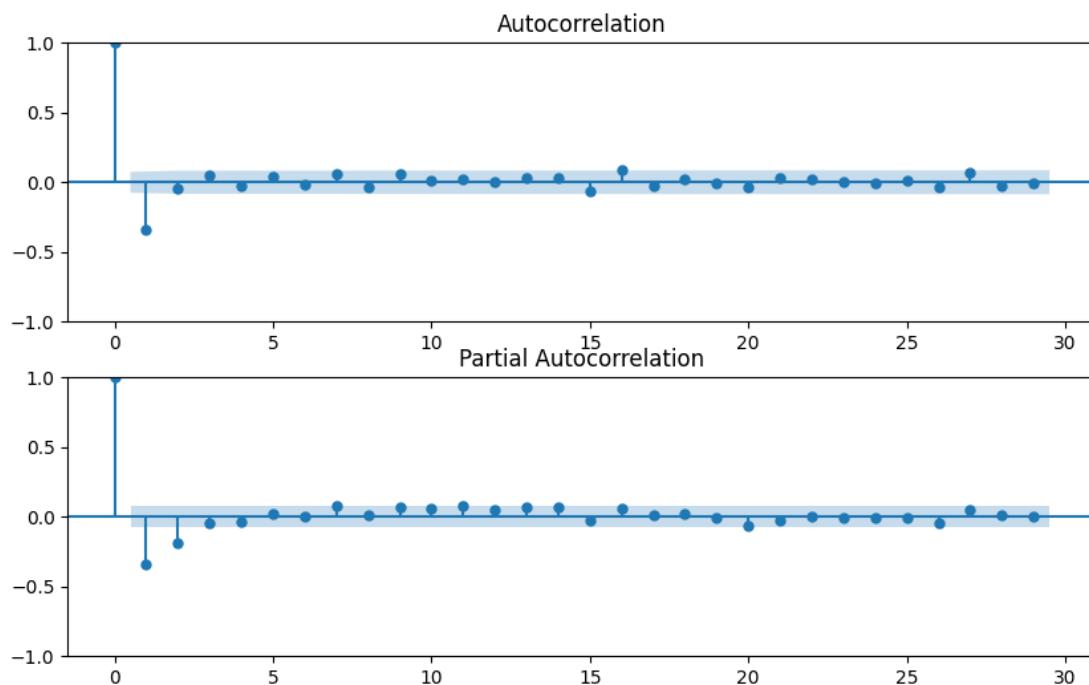
Pada beras medium setelah dilakukan transformasi dan *differencing* pertama, diperoleh p-value sebesar 3.7866e-05, yang juga lebih kecil dari 0,05. Nilai ADF *test statistic* yang dihasilkan juga lebih rendah dari semua nilai *critical values*. Maka dapat disimpulkan data harga beras medium juga telah stasioner.

4.4.3 Identifikasi parameter model SARIMA

Setelah data telah stasioner, langkah berikutnya adalah menentukan parameter model menggunakan grafik ACF dan PACF. Analisis grafik ACF dan PACF ini akan membantu dalam pemilihan parameter terbaik dengan menggunakan fungsi *auto-arima*.



Gambar 4.4.3.1 Grafik ACF dan PACF (beras premium)



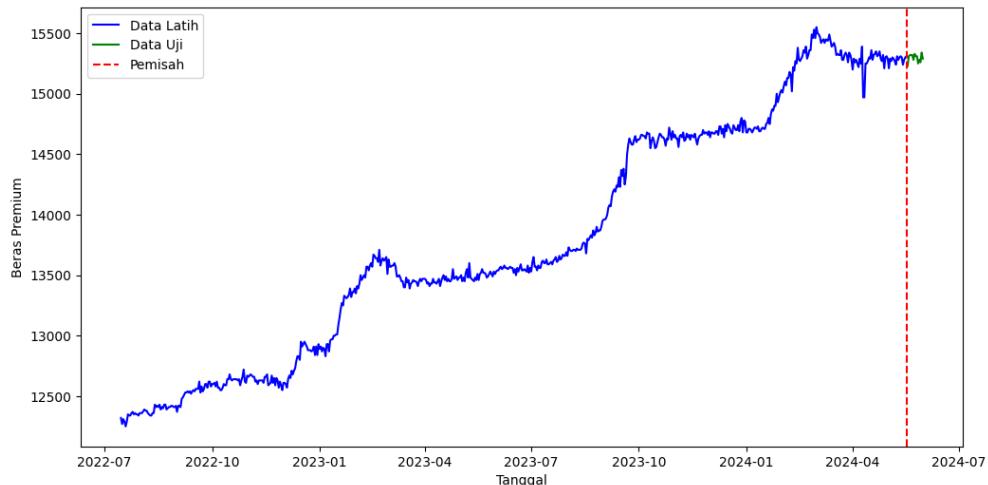
Gambar 4.4.3.2 Grafik ACF dan PACF (beras medium)

Berdasarkan plot pada gambar 4.4.3.1, grafik ACF memperlihatkan *cut off* setelah *lag* 1, maka orde $q(1)$ dijadikan orde q maksimal pada *auto-arima*. Grafik PACF menunjukkan *cut off* setelah *lag* 2, maka orde $p(2)$ dijadikan orde p maksimal. Untuk orde *seasonal* nilai maksimal masing-masing adalah $P(3)$ dan $Q(3)$.

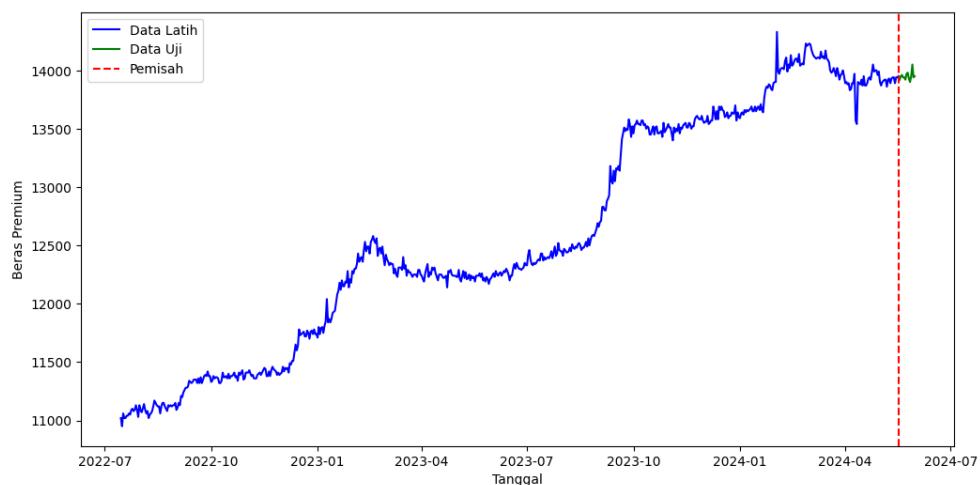
Berdasarkan plot pada gambar 4.4.3.2, grafik ACF memperlihatkan *cut off* setelah *lag* 1, sehingga orde $q(1)$ dijadikan orde q maksimal pada *auto-arima*. Grafik PACF menunjukkan *cut off* setelah *lag* 2, maka orde $p(2)$ dijadikan orde p maksimal. Untuk orde *seasonal* nilai maksimal masing-masing adalah $P(3)$ dan $Q(3)$.

4.4.4 Pemilihan model terbaik

Sebelum masuk ke tahap pemodelan, dilakukan pembagian data menjadi latih dan uji. Data latih yang digunakan dimulai dari tanggal 15 Juli 2022 - 17 Mei 2024. Sedangkan untuk data uji atau *out of sample* akan berjumlah 14 hari dimulai dari 18 Mei - 31 Mei 2024. Data uji akan digunakan sebagai evaluasi hasil prediksi model SARIMA. Visualiasi pembagian data untuk model SARIMA dapat diperhatikan pada gambar 4.4.4.1 dan 4.4.4.2.



Gambar 4.4.4.1 Visualisasi pembagian data model SARIMA (BP)



Gambar 4.4.4.2 Visualisasi pembagian data model SARIMA (BM)

Selanjutnya dilakukan pemilihan model dengan menggunakan fungsi *auto-arima* yang disediakan pustaka pmdarima. Dengan orde maksimal yang telah diperoleh pada proses analisis plot ACF dan PACF, orde tersebut akan dimasukkan kedalam parameter fungsi *auto-arima*. Parameter AR dan MA (termasuk musiman) akan dimulai dari 0. Untuk nilai maksimal parameter AR dan MA diatur menjadi 2 dan 1 untuk komponen non-musiman. Untuk komponen non-musian parameter maksimal AR dan MA diatur menjadi 3 dan 3.

```
Best model: ARIMA(2,1,0)(3,0,0)[7]
Total fit time: 97.093 seconds
SARIMAX Results
=====
Dep. Variable: y No. Observations: 673
Model: SARIMAX(2, 1, 0)x(3, 0, 0, 7) Log Likelihood: 5922.607
Date: Thu, 03 Oct 2024 AIC: -11833.214
Time: 16:04:36 BIC: -11806.153
Sample: 07-15-2022 HQIC: -11822.734
- 05-17-2024
Covariance Type: opg
=====
```

Gambar 4.4.4.3 Hasil *auto-arima* beras premium

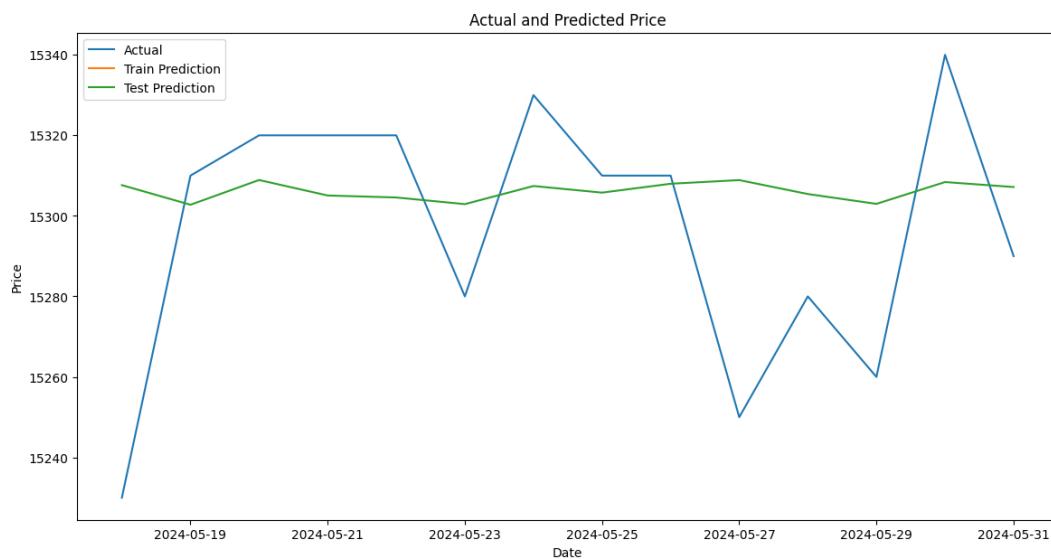
```
Best model: ARIMA(0,1,1)(1,0,1)[7]
Total fit time: 141.939 seconds
SARIMAX Results
=====
Dep. Variable: y No. Observations: 673
Model: SARIMAX(0, 1, 1)x(1, 0, 1, 7) Log Likelihood: 570.478
Date: Thu, 03 Oct 2024 AIC: -1132.957
Time: 14:43:05 BIC: -1114.916
Sample: 07-15-2022 HQIC: -1125.970
- 05-17-2024
Covariance Type: opg
=====
```

Gambar 4.4.4.4 Hasil *auto-arima* beras medium

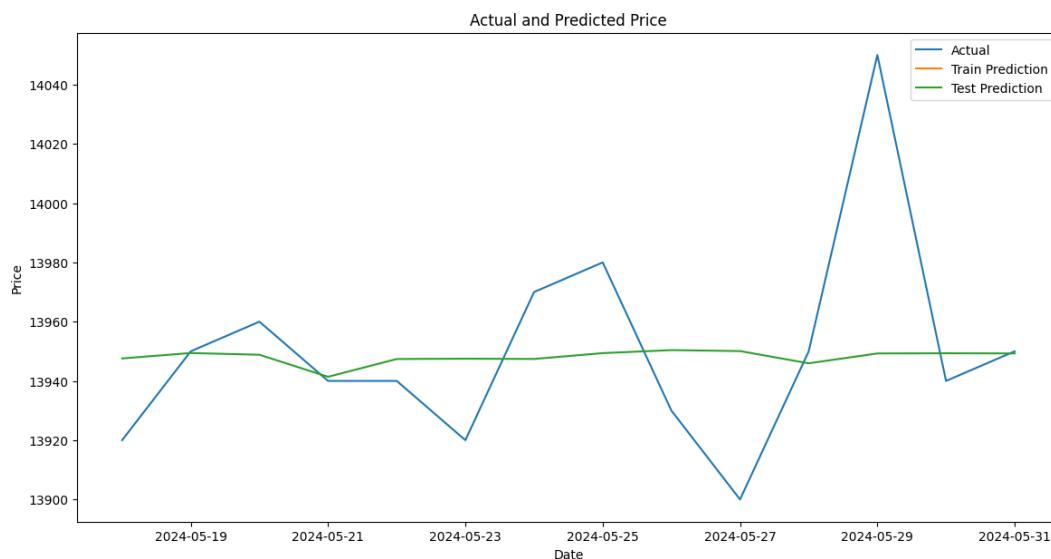
Selanjutnya *auto-arima* akan melatih beberapa model dengan menggunakan semua kombinasi parameter yang mungkin. Dari seluruh model yang telah dilatih, *auto-arima* akan memilih model dengan nilai AIC terendah. Berdasarkan gambar 4.4.4.1 dan 4.4.4.2, kombinasi parameter terbaik untuk memprediksi data harga beras premium adalah SARIMA(2, 1, 0)(3, 0, 0)⁷ dan untuk beras medium adalah SARIMA(0,1,1)(1,0,1)⁷.

4.4.5 Prediksi dan Evaluasi

Prediksi harga beras premium dan medium untuk 14 hari ke depan dilakukan menggunakan model terbaik yang telah diidentifikasi untuk kedua jenis beras. Hasil prediksi ini digunakan untuk mengevaluasi performa model untuk memprediksi data yang belum pernah digunakan dalam pelatihan. Untuk mendapatkan hasil akhir, hasil prediksi perlu dikembalikan ke skala aslinya. Proses transformasi balik ini melibatkan fungsi `inv_boxcox()` yang disediakan pustaka `scipy.special`, dengan parameter `lambda` diisi dengan nilai `lambda` yang diperoleh dari proses transformasi Box-Cox sebelumnya.



Gambar 4.4.5.1 Grafik perbandingan aktual vs prediksi harga BP (SARIMA)



Gambar 4.4.5.2 Grafik perbandingan aktual vs prediksi harga BM (SARIMA)

Berdasarkan gambar 4.4.5.1 hasil prediksi harga beras premium pada data uji sudah cukup baik dengan berada disekitar harga aktual, namun pola prediksi tampaknya lebih stabil dibandingkan harga aktual. Hasil prediksi cenderung lebih rata dan tidak dapat menangkap fluktuasi yang tajam pada harga aktual. Hal yang sama juga terlihat pada hasil prediksi harga beras medium pada gambar 4.4.5.2, prediksi berada di sekitar harga aktual, namun polanya cenderung lebih stabil. Hasil prediksi kedua jenis beras dapat dilihat pada tabel 4.4.5.1.

Tabel 4.4.5.1 Hasil prediksi harga beras dengan model SARIMA

Tanggal	Beras Premium		Beras Medium	
	Aktual	Prediksi	Aktual	Prediksi
2024-05-18	15230	15307,63	13920	13947,59
2024-05-19	15310	15302,75	13950	13949,42
2024-05-20	15320	15308,90	13960	13948,83
2024-05-21	15320	15305,05	13940	13941,39
2024-05-22	15320	15304,56	13940	13947,39
2024-05-23	15280	15302,91	13920	13947,50
2024-05-24	15330	15307,41	13970	13947,41
2024-05-25	15310	15305,77	13980	13949,38
2024-05-26	15310	15307,97	13930	13950,40
2024-05-27	15250	15308,88	13900	13950,07
2024-05-28	15280	15305,42	13950	13945,94
2024-05-29	15260	15302,98	14050	13949,27
2024-05-30	15340	15308,39	13940	13949,33
2024-05-31	15290	15307,16	13950	13949,28

Berikut ini contoh perhitungan prediksi beras premium menggunakan model SARIMA(2,1,0)(3,0,0)⁷, dengan $\Phi_1 = 0,0522$, $\Phi_2 = 0,0423$, $\Phi_3 = 0,0599$, $\phi_1 = -0,4118$, $\phi_2 = -0,1103$.

$$\Phi_p(B^s)\phi_p(B)(1 - B^s)^D(1 - B)^d y_t = \Theta_Q(B^s)\theta_q(B)\varepsilon_t$$

$$(1 - \Phi_1 B^7 - \Phi_2 B^{14} - \Phi_3 B^{21})(1 - \phi_1 B - \phi_2 B^2)(1 - B)_{y_t} = \varepsilon_t$$

$$(1 - \Phi_1 B^7 - \Phi_2 B^{14} - \Phi_3 B^{21})(1 - \phi_1 B - \phi_2 B^2)\nabla_{y_t} = \varepsilon_t$$

$$\begin{aligned}
& (1 - \phi_1 B - \phi_2 B^2 - \Phi_1 B^7 + \Phi_1 \phi_1 B^8 + \Phi_1 \phi_2 B^9 - \Phi_2 B^{14} + \Phi_2 \phi_1 B^{15} + \\
& \Phi_2 \phi_2 B^{16} - \Phi_3 B^{21} + \Phi_3 \phi_1 B^{22} + \Phi_3 \phi_2 B^{23}) \nabla_{y_t} = \varepsilon_t \\
& \nabla_{y_t} - \phi_1 \nabla_{y_{t-1}} - \phi_2 \nabla_{y_{t-2}} - \Phi_1 \nabla_{y_{t-7}} + \Phi_1 \phi_1 \nabla_{y_{t-8}} + \Phi_1 \phi_2 \nabla_{y_{t-9}} - \Phi_2 \nabla_{y_{t-14}} + \\
& \Phi_2 \phi_1 \nabla_{y_{t-15}} + \Phi_2 \phi_2 \nabla_{y_{t-16}} - \Phi_3 \nabla_{y_{t-21}} + \Phi_3 \phi_1 \nabla_{y_{t-22}} + \Phi_3 \phi_2 \nabla_{y_{t-23}} = \varepsilon_t \\
& \nabla_{y_t} = \phi_1 \nabla_{y_{t-1}} + \phi_2 \nabla_{y_{t-2}} + \Phi_1 \nabla_{y_{t-7}} - \Phi_1 \phi_1 \nabla_{y_{t-8}} - \Phi_1 \phi_2 \nabla_{y_{t-9}} + \Phi_2 \nabla_{y_{t-14}} - \\
& \Phi_2 \phi_1 \nabla_{y_{t-15}} - \Phi_2 \phi_2 \nabla_{y_{t-16}} + \Phi_3 \nabla_{y_{t-21}} - \Phi_3 \phi_1 \nabla_{y_{t-22}} - \Phi_3 \phi_2 \nabla_{y_{t-23}} + \varepsilon_t \\
& \nabla_{y_t} = (-0,4118 * 0,0000007) + (-0,1103 * 0,0000015) + (0,0522 * 0,0000015) - \\
& (0,0522 * -0,4118 * -0,000022) - (0,0522 * -0,1103 * 0,000051) + \\
& (0,0423 * -0,000015) - (0,0423 * -0,4118 * 0,000059) - (0,0423 * \\
& -0,1103 * -0,000059) + (0,0599 * 0,000037) - (0,0599 * -0,4118 * \\
& -0,000015) - (0,0599 * -0,1103 * -0,000044) + \varepsilon_t \\
& \nabla_{y_t} = (-0,0000028) + (-0,0000016) + (0,0000007) - (0,0000004) - \\
& (-0,0000002) + (0,0000006) - (0,0000010) - (0,0000002) + \\
& (0,0000022) - (0,0000003) - (-0,0000002) + \varepsilon_t \\
& \nabla_{y_t} = -0,000002 + \varepsilon_t \\
& \widehat{\nabla_{y_t}} = \widehat{y_t} - y_{t-d} \\
& \widehat{y_t} = y_{t-d} + \widehat{\nabla_{y_t}} \\
& \widehat{y_t} = 2,121275 + (-0,000002) \\
& \widehat{y_t} = 2,121273
\end{aligned}$$

Selanjutnya hasil diatas dikembalikan skalanya dengan menggunakan fungsi `inv_boxcox()` sehingga didapatkan hasil 15307.

Hasil prediksi kedua model selanjutnya di evaluasi menggunakan metrik evaluasi MAE dan MAPE untuk mengukur kemampuan model dalam melakukan prediksi harga di masa depan.

1. Beras Premium

$$MAE = \frac{1}{n} \sum_{t=1}^n |X_t - \hat{X}_t|$$

$$\begin{aligned}
MAE &= \frac{(|15230 - 15307,63| + |15310 - 15302,75| + \dots + |15290 - 15307,16|)}{14} \\
MAE &= \frac{(77,63 + 7,25 + \dots + 17,16)}{14} \\
MAE &= \frac{354,180}{14} \\
MAE &= 25,296
\end{aligned}$$

$$\begin{aligned}
MAPE &= \frac{1}{n} \sum_{i=1}^n \left| \frac{X_i - \hat{X}_i}{X_i} \right| * 100 \\
&\quad \left(\left| \frac{15230 - 15307,63}{15230} \right| * 100 \right) + \left(\left| \frac{15310 - 15302,75}{15310} \right| * 100 \right) + \dots + \\
MAPE &= \frac{\left(\left| \frac{15290 - 15307,16}{15290} \right| * 100 \right)}{14} \\
MAPE &= \frac{(0,509718 + 0,047355 + \dots + 0,112230)}{14} \\
MAPE &= \frac{2,318}{14} \\
MAPE &= 0,165
\end{aligned}$$

2. Beras Medium

$$\begin{aligned}
MAE &= \frac{1}{n} \sum_{t=1}^n |X_t - \hat{X}_t| \\
&\quad \left(|13920 - 13947,59| + |13950 - 13949,42| + \dots + |13950 - 13949,28| \right) \\
MAE &= \frac{(27,59 + 0,58 + \dots + 0,72)}{14} \\
MAE &= \frac{314,140}{14} \\
MAE &= 22,438
\end{aligned}$$

$$\begin{aligned}
MAPE &= \frac{1}{n} \sum_{i=1}^n \left| \frac{X_i - \hat{X}_i}{X_i} \right| * 100 \\
&\quad \left(\left| \frac{13920 - 13947,59}{13920} \right| * 100 \right) + \left(\left| \frac{13950 - 13949,42}{13950} \right| * 100 \right) + \dots + \\
MAPE &= \frac{\left(\left| \frac{13950 - 13949,28}{13950} \right| * 100 \right)}{14} \\
MAPE &= \frac{(0,198204 + 0,004158 + \dots + 0,005161)}{14}
\end{aligned}$$

$$MAPE = \frac{2,248}{14}$$

$$MAPE = 0,160$$

Tabel 4.4.5.2 Metrik evaluasi prediksi harga beras

Jenis Beras	MAE	MAPE (%)
Premium	25,296	0,165
Medium	22,438	0,160

Jika dilihat nilai MAE dan MAPE keduanya tampaknya sangat baik dengan nilai MAPE keduanya dibawah 1%. Namun jika dilihat dari grafik perbandingan prediksi dan harga aktual nampaknya kedua model tidak bisa menangkap fluktuasi harga aktual beras premium dan medium pada data uji dengan cukup baik.

4.5 Implementasi Model LSTM

Model kedua yang digunakan untuk memprediksi harga beras premium dan beras medium adalah model LSTM. Berikut ini tahapan implementasi model LSTM

4.5.1 Pra-pemrosesan data untuk model LSTM

Sebelum melatih model LSTM terdapat beberapa tahapan pra-pemrosesan yang perlu dilakukan. Tahap awal adalah melakukan normalisasi data dengan menerapkan teknik *min-max scaling* dengan menggunakan rentang skala 0,0 hingga 1,0. Proses ini melibatkan fungsi MinMaxScaler yang disediakan pustaka sklearn. Berikut adalah beberapa hasil perubahan data setelah dilakukan *scaling*.

Tabel 4.5.1.1 Data setelah normalisasi

Index	Beras Premium	Beras Medium
0	0,021212	0,020710
1	0,006061	0,000000
2	0,018182	0,032544
3	0,012121	0,020710
4	0,000000	0,020710

Pada tabel 4.5.1.1 data dirubah skalanya kedalam rentang 0,0 sampai 1,0. Data dengan nilai paling kecil dirubah nilainya menjadi 0,0 dan data terbesar menjadi 1,0.

Data diantara nilai terbesar dan terkecil dirubah nilainya diantara rentang tersebut. Tahap berikutnya adalah membagi data menjadi data *input* dan *output*. Proses ini melibatkan metode *sliding window*. Untuk *window size* yang digunakan adalah 5 dan jumlah *output* adalah 14, dimana masing-masing kolom *output* akan menjadi target untuk masing-masing model LSTM yang akan dibuat. Berikut adalah beberapa hasil pembagian data *input* dan *output* pada beras premium.

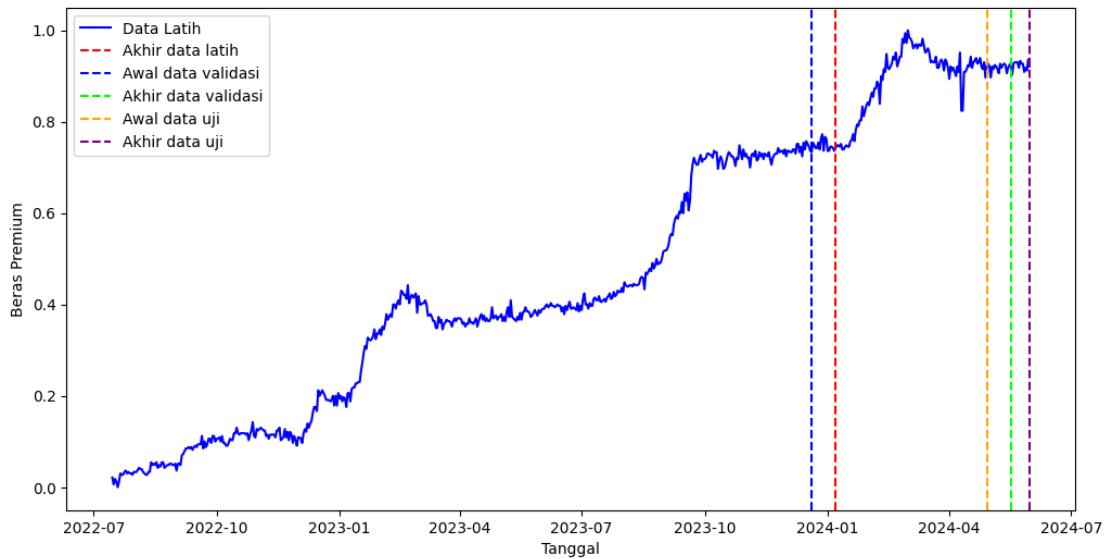
Tabel 4.5.1.2 Data input untuk beras premium

Waktu-5	Waktu-4	Waktu-3	Waktu-2	Waktu-1
0,02121212	0,00606061	0,01818182	0,01212121	0
0,00606061	0,01818182	0,01212121	0	0,01212121
0,01818182	0,01212121	0	0,01212121	0,03030303
0,01212121	0	0,01212121	0,03030303	0,02727273
0	0,01212121	0,03030303	0,02727273	0,02727273

Tabel 4.5.1.3 Data output untuk beras premium

Model1	Model2	Model3	...	Model13	Model14
0,01212121	0,03030303	0,02727273	...	0,03333333	0,03333333
0,03030303	0,02727273	0,02727273	...	0,03333333	0,03787879
0,02727273	0,02727273	0,03333333	...	0,03787879	0,04242424
0,02727273	0,03333333	0,03636364	...	0,04242424	0,03939394
0,03333333	0,03636364	0,03030303	...	0,03939394	0,03939394

Pada tabel 4.5.1.2, sejumlah 5 data di masa lalu digunakan sebagai data *input* untuk 14 model, di mana setiap model akan dilatih untuk memprediksi *output* pada tabel 4.5.1.3. Data *input* untuk pelatihan model LSTM harus berbentuk 3 dimensi, sehingga data *input* perlu untuk dilakukan *reshaping*. Setelah dilakukan *reshaping*, data *input* akan memiliki bentuk (669, 5, 1). Setelah itu dilakukan pembagian data latih, data validasi, dan data uji. Empat belas baris terakhir dipakai untuk data uji, sementara sisanya dipakai untuk data latih. Data validasi akan diambil dari data latih dengan rasio 80:20. Visualisasi pembagian data ditampilkan pada gambar 4.5.1.1.



Gambar 4.5.1.1 Visualisasi pembagian data LSTM

4.5.2 Pelatihan model LSTM

Sebelum masuk ke tahap pelatihan model LSTM, penting untuk menentukan *hyperparameter* model. Proses penentuan ini melibatkan *hyperparameter tuning*, dengan tujuan untuk mendapatkan kombinasi parameter terbaik guna menghasilkan prediksi yang akurat. Model akan dipilih berdasarkan nilai *validation loss* yang paling rendah. Tujuan penggunaan kriteria ini adalah untuk mendapatkan model yang tidak hanya baik dalam memprediksi data latih, tetapi juga pada data yang belum pernah dilihat. Proses ini melibatkan penggunaan pustaka keras-tuner. Untuk mempersingkat proses pencarian, metode pencarian yang digunakan adalah *Random Search*. Berikut ini adalah daftar parameter yang akan di uji coba.

Tabel 4.5.2.1 Daftar pengujian parameter LSTM

Parameter	Variasi
LSTM	16, 32, 64
Dropout	0,1, 0,2, 0,3, 0,4
LSTM_1	16, 32, 64
Dropout_1	0,1, 0,2, 0,3, 0,4
Learning_rate	0,005, 0,0005

Berdasarkan tabel 4.5.2.1 arsitektur model yang dibangun memiliki dua layer LSTM dengan penambahan layer *dropout* setelah keduanya. Tujuan penambahan layer

dropout setelah layer LSTM adalah untuk mencegah model menjadi *overfitting*. Setelah itu proses pencarian dilakukan dengan menggunakan *epochs* sebanyak 100, *batch size* 128, dan *early stopping* dengan nilai *patience* adalah 20. Penggunaan *early stopping* bertujuan untuk menghentikan proses pelatihan jika model tidak menunjukkan perbaikan pada *validation loss* selama 20 epoch agar waktu pelatihan lebih efisien. Daftar parameter yang didapat dari proses *hyperparameter tuning* ditampilkan pada tabel 4.5.2.2.

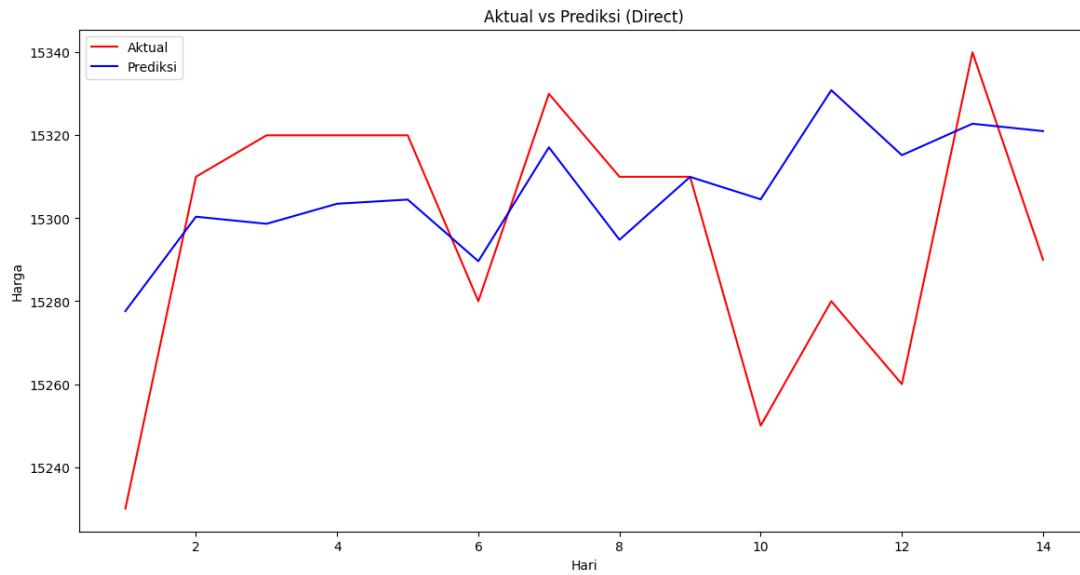
Tabel 4.5.2.2 Kombinasi parameter terbaik untuk prediksi harga beras

Parameter	Nilai	
	Beras Premium	Beras Medium
LSTM	64 units	96 units
Dropout	0,2	0,1
LSTM_1	32 units	32 units
Dropout_1	0,4	0,1
Learning_rate	0,005	0,005

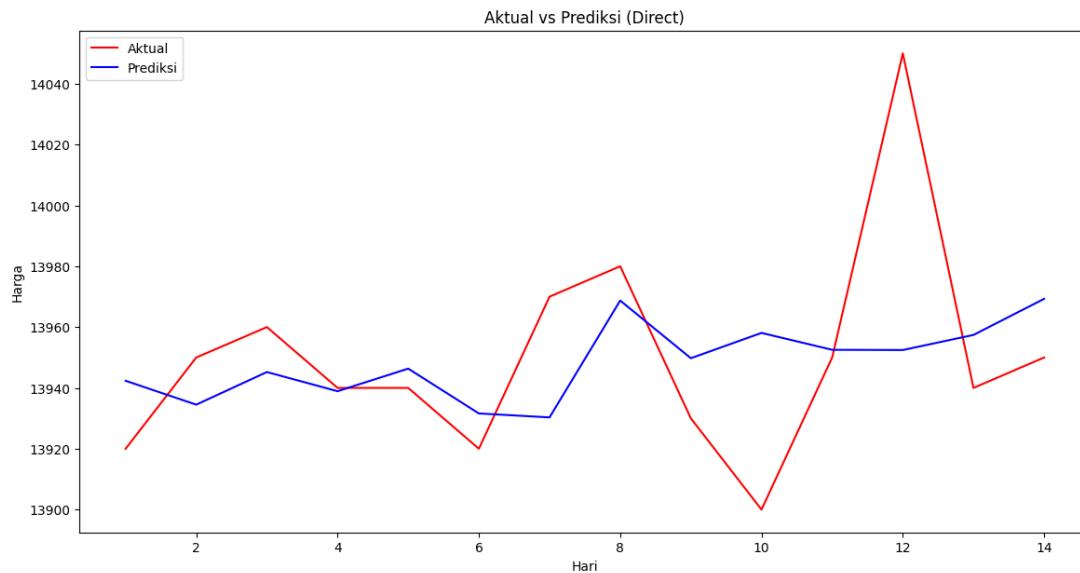
Setelah proses pencarian *hyperparamter* selesai, daftar *hyperparameter* terbaik kemudian diambil dengan menggunakan fungsi `get_best_hyperparameters()`. Selanjutnya dilakukan pelatihan sebanyak 14 model dengan *hyperparameter* terbaik yang didapat sebelumnya. Untuk parameter pelatihan yang digunakan adalah kombinasi parameter terbaik yang didapat dari proses *hyperparameter tuning*.

4.5.3 Pengujian dan evaluasi

Pengujian model dilakukan dengan memprediksi 14 hari terakhir pada dataset. Untuk mendapatkan hasil akhir, hasil prediksi perlu di denormalisasi untuk mengembalikannya ke skala aslinya. Proses ini melibatkan penggunaan fungsi `inverse_transform()` yang terdapat pada `MinMaxScaller`. Berikut ini plot perbandingan harga aktual dan harga prediksi.



Gambar 4.5.3.1 Grafik perbandingan aktual vs prediksi harga BP (LSTM)



Gambar 4.5.3.2 Grafik perbandingan aktual vs prediksi harga BM (LSTM)

Berdasarkan gambar 4.5.3.1, garis berwarna merah merupakan harga aktual dan garis berwarna biru adalah harga prediksi. Pada hari pertama tampaknya terdapat perbedaan prediksi yang cukup tinggi dengan harga aktual, namun pada hari berikutnya akurasi prediksi membaik sampai hari ke-9. Pada hari ke-10 – ke 12 terdapat penurunan ketepatan prediksi, dimana pada periode itu terdapat penurunan yang cukup tajam pada harga aktual.

Pada gambar 4.5.3.2, harga beras medium tampaknya berfluktuasi lebih kecil. Hasil prediksi model menunjukkan perbedaan yang kecil di banyak titik prediksi. Namun pada hari ke-10 dan ke-12 terjadi penurunan ketepatan prediksi, dimana pada periode tersebut terdapat penurunan dan kenaikan yang cukup tajam pada harga aktual yang tidak bisa diprediksi dengan baik. Hasil prediksi kedua model ditampilkan pada tabel 4.5.3.1.

Tabel 4.5.3.1 Prediksi 14 hari kedepan dengan model LSTM

Tanggal	Beras Premium		Beras Medium	
	Aktual	Prediksi	Aktual	Prediksi
2024-05-18	15230	15277.61	13920	13942.34
2024-05-19	15310	15300.37	13950	13934.49
2024-05-20	15320	15298.66	13960	13945.22
2024-05-21	15320	15303.50	13940	13938.91
2024-05-22	15320	15304.50	13940	13946.34
2024-05-23	15280	15289.64	13920	13931.60
2024-05-24	15330	15317.10	13970	13930.31
2024-05-25	15310	15294.81	13980	13968.72
2024-05-26	15310	15309.97	13930	13949.76
2024-05-27	15250	15304.56	13900	13958.08
2024-05-28	15280	15330.87	13950	13952.53
2024-05-29	15260	15315.20	14050	13952.45
2024-05-30	15340	15322.79	13940	13957.42
2024-05-31	15290	15321.00	13950	13969.29

Berikut ini contoh perhitungan prediksi menggunakan model LSTM, dengan daftar nilai variabelnya dapat diperhatikan pada tabel 4.5.1.

Tabel 4.5.1 Nilai variabel untuk contoh perhitungan LSTM

Variabel	Nilai
x_t	0,92424244
h_{t-1}	[0, 0, 0, ..., 0, 0, 0] (Total=64)
C_{t-1}	0
W_i	[-0,00152381, 0,15107544, ..., -0,0587961, -0,00281798] (Total=64), -0,05087651]

Variabel	Nilai
W_f	[[-0,06133534, 0,0472329, ..., 0,14783016, -0,03148066] (Total=64), 0,17890172]
W_c	[[-0,00771869, 0,05856314, ..., -0,03669531, -0,01403097] (Total=64), -0,0974368]
W_o	[[-0,16418639, 0,11448701, ..., 0,10359919, -0,02855348] (Total=64), -0,08475205]
b_i	0,015488874
b_f	1,0174892
b_c	-0,009995385
b_o	0,015743015

$$\begin{aligned}
 f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\
 &= \sigma(([-0,06133534, 0,0472329, \dots, 0,14783016, -0,03148066] \cdot \\
 &\quad [0, 0, 0, \dots, 0, 0, 0]) + (0,17890172 \cdot 0,92424244) + 1,0174892) \\
 &= \sigma(0 + (0,16534856) + 1,0174892) \\
 &= \sigma(1,1828377) \\
 &= 0,7654576588101163
 \end{aligned}$$

$$\begin{aligned}
 i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\
 &= \sigma(([-0,00152381, 0,15107544, \dots, -0,0587961, -0,00281798] \cdot \\
 &\quad [0, 0, 0, \dots, 0, 0, 0]) + (-0,05087651 \cdot 0,92424244) + 0,015488874) \\
 &= \sigma(0 + -0,04702223 + 0,015488874) \\
 &= \sigma(-0,031533357) \\
 &= 0,4921173073818347
 \end{aligned}$$

$$\begin{aligned}
 \tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) \\
 &= \tanh(([-0,00771869, 0,05856314, \dots, -0,03669531, -0,01403097] \cdot \\
 &\quad [0, 0, 0, \dots, 0, 0, 0]) + (-0,0974368 \cdot 0,92424244) + -0,009995385)
 \end{aligned}$$

$$\begin{aligned}
&= \tanh(0 + (-0,09005523) + -0,009995385) \\
&= \tanh(-0,10005061) \\
&= -0,099718094 \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
&= (0,7654576588101163 * 0) + (0,4921173073818347 * -0,099718094) \\
&= 0 + (-0,04907299991652868) \\
&= -0,04907299991652868 \\
o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\
&= \sigma((-0,16418639, 0,11448701, \dots, 0,10359919, -0,02855348] \cdot \\
&\quad [0, 0, 0, \dots, 0, 0, 0]) + (-0,08475205 \cdot 0,92424244) + 0,015743015) \\
&= \sigma(0 + -0,07833144 + 0,015743015) \\
&= \sigma(-0,06258842) \\
&= 0,48435798204575603 \\
h_t &= o_t * \tanh(C_t) \\
&= 0,48435798204575603 * \tanh(-0,04907299991652868) \\
&= 0,48435798204575603 * -0,04903364 \\
&= -0,023749836
\end{aligned}$$

Nilai h_t dan C_t yang didapat kemudian diberikan sebagai *input* untuk proses perhitungan nilai h_t dan C_t untuk timestep selanjutnya.

Tabel 4.5.3.2 Metrik evaluasi model LSTM

Jenis Beras	MAE	MAPE (%)
Beras Premium	25,513	0,166
Beras Medium	24,090	0,172

Berdasarkan tabel 4.5.3.2, kedua model LSTM menunjukkan tingkat *error* sangat kecil. Keduanya memiliki nilai MAPE dibawah 1%, yaitu 0.166% untuk beras premium dan 0.172% pada beras medium. Nilai MAPE ini mengindikasikan bahwa kedua model ini memiliki kemampuan prediksi yang sangat baik.

4.6 Implementasi Model *Hybrid* SARIMA-LSTM

Model ketiga yang akan dibuat untuk memprediksi harga beras adalah model *Hybrid* SARIMA-LSTM. Model ini dibuat dengan tujuan untuk memberikan hasil yang paling optimal, dimana kelebihan dari kedua model dapat dimanfaatkan untuk memberikan hasil prediksi yang lebih baik. Berikut ini adalah tahapan implementasi model *Hybrid* SARIMA-LSTM.

4.6.1 Pengambilan residual dari model SARIMA

Model SARIMA dengan parameter terbaik yang telah didapat sebelumnya akan digunakan kembali untuk memprediksi data latih dari pemodelan SARIMA sebelumnya. Nilai *residual* dari model ini kemudian diambil dengan menghitung selisih antara harga aktual dan harga prediksi. *Residual* ini kemudian akan digunakan sebagai data latih untuk model LSTM. Beberapa hasil perhitungan *residual* model SARIMA untuk beras premium dan beras medium ditampilkan pada tabel 4.6.1.1.

Tabel 4.6.1.1 Residual model SARIMA

Tanggal	Residual	
	Beras Premium	Beras Medium
2022-07-16	-50,000000	-69,999862
2022-07-17	19,238958	85,330906
2022-07-18	-8,987431	-5,551812
2022-07-19	-43,594360	-2,283881
2022-07-20	21,229332	19,057184

4.6.2 Pelatihan model LSTM

Pada tahap ini, model LSTM akan dilatih untuk memprediksi *residual* yang didapat dari model SARIMA. Sebelum dilatih data *residual* akan dilakukan pra-pemrosesan data, yaitu normalisasi dan pembentukan data input dan output. Model LSTM yang dilatih berjumlah tiga buah dengan rasio pembagian data latih dan validasi, yaitu 60:40, 70:30, dan 80:40. Untuk *window size* yang digunakan adalah 5 dan jumlah kolom *output* adalah

14. Setelah itu, dilakukan pencarian *hyperparameter* terbaik untuk memprediksi data *residual*.

Untuk daftar parameter yang akan diuji, daftarnya sama dengan yang ada pada proses *hyperparameter tuning* untuk model LSTM sebelumnya. Proses pencarian ini menggunakan *epochs* sebesar 1000, dan *batch size* 32. Hasil *hyperparameter tuning* untuk ketiga model LSTM untuk beras premium dan beras medium ditampilkan pada tabel 4.6.2.1, 4.6.2.2, dan 4.6.2.3.

Tabel 4.6.2.1 Kombinasi parameter terbaik untuk *training set* 80%

Parameter	Nilai	
	Beras Premium	Beras Medium
LSTM	64 units	64 units
Dropout	0,3	0,3
LSTM_1	32 units	64 units
Dropout_1	0,1	0,3
Learning_rate	0,005	0,005

Tabel 4.6.2.2 Kombinasi parameter terbaik untuk *training set* 70%

Parameter	Nilai	
	Beras Premium	Beras Medium
LSTM	64 units	64 units
Dropout	0,3	0,3
LSTM_1	32 units	64 units
Dropout_1	0,1	0,3
Learning_rate	0,005	0,005

Tabel 4.6.2.3 Kombinasi parameter terbaik untuk *training set* 60%

Parameter	Nilai	
	Beras Premium	Beras Medium
LSTM	64 units	64 units
Dropout	0,3	0,3
LSTM_1	32 units	64 units
Dropout_1	0,1	0,3
Learning_rate	0,005	0,005

Dapat dilihat dari ketiga tabel diatas kombinasi parameter terbaik yang didapat dari proses *hyperparameter tuning* sama untuk ketiga model LSTM pada kedua jenis beras. Tahap berikutnya adalah melakukan pelatihan model LSTM dengan menggunakan *hyperparameter* terbaik yang didapat sebelumnya. Parameter pelatihan model sama dengan parameter yang digunakan pada proses *hyperparameter tuning*. Seluruh model kemudian dimasukkan kedalam *list* agar seluruh model tersimpan dalam satu *variable* sehingga mempermudah penggunaan model.

4.6.3 Pengujian dan evaluasi

Pada tahap ini, model *Hybrid* SARIMA-LSTM digunakan untuk memprediksi 14 data terakhir dalam dataset. Proses dimulai dengan model SARIMA yang berperan dalam memprediksi nilai harga beras. Setelah itu, *residual* dari prediksi SARIMA dihitung sebagai selisih antara nilai aktual dan hasil prediksi SARIMA. *Residual* ini selanjutnya digunakan sebagai data input untuk model LSTM.

Tahap berikutnya adalah melakukan prediksi *residual* dengan model LSTM. Hasil prediksi LSTM kemudian dilakukan *denormalisasi* untuk mengembalikannya ke skala aslinya. Setelah kedua model (SARIMA dan LSTM) melakukan prediksi, langkah berikutnya adalah menggabungkan hasilnya untuk mendapatkan prediksi akhir.

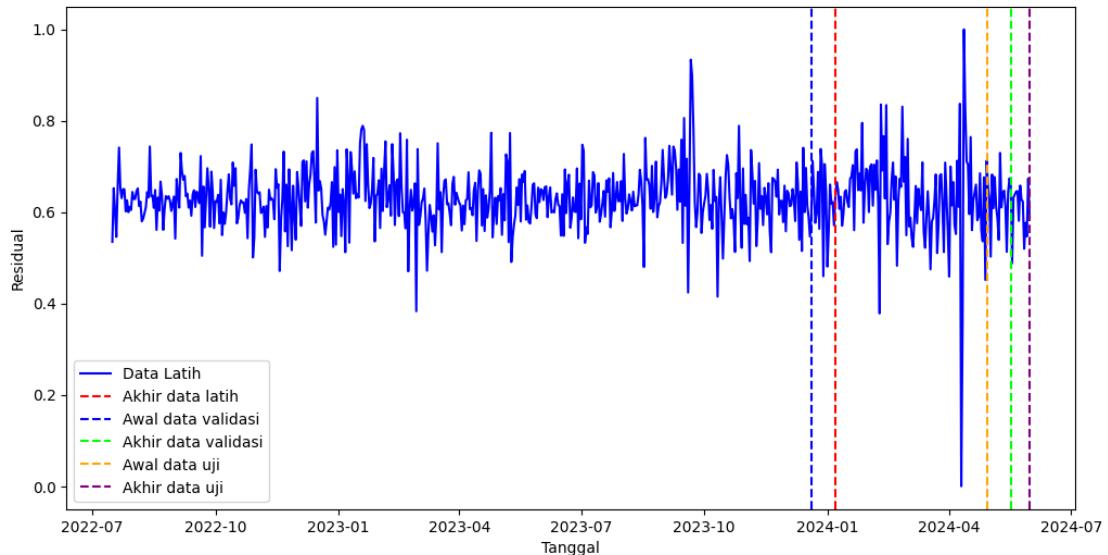
Sebelum penggabungan, hasil prediksi model SARIMA harus diubah terlebih dahulu kedalam bentuk data *input* dan *output* seperti pada model LSTM, dimana hanya data *output* yang akan digunakan. Setelah bentuk kedua data hasil prediksi sama, penggabungan dilakukan untuk mendapatkan hasil akhir prediksi model *Hybrid* SARIMA-LSTM. Berikut ini adalah perbandingan performa ketiga model *Hybrid* dalam memprediksi 14 hari terakhir pada dataset.

Tabel 4.6.3.1 Pengujian model *Hybrid* SARIMA-LSTM

<i>Training Set LSTM</i>	Beras Premium		Beras Medium	
	MAE	MAPE (%)	MAE	MAPE (%)
80%	24,168	0,158	21,680	0,155
70%	29,610	0,193	23,929	0,171
60%	27,548	0,180	23,607	0,169

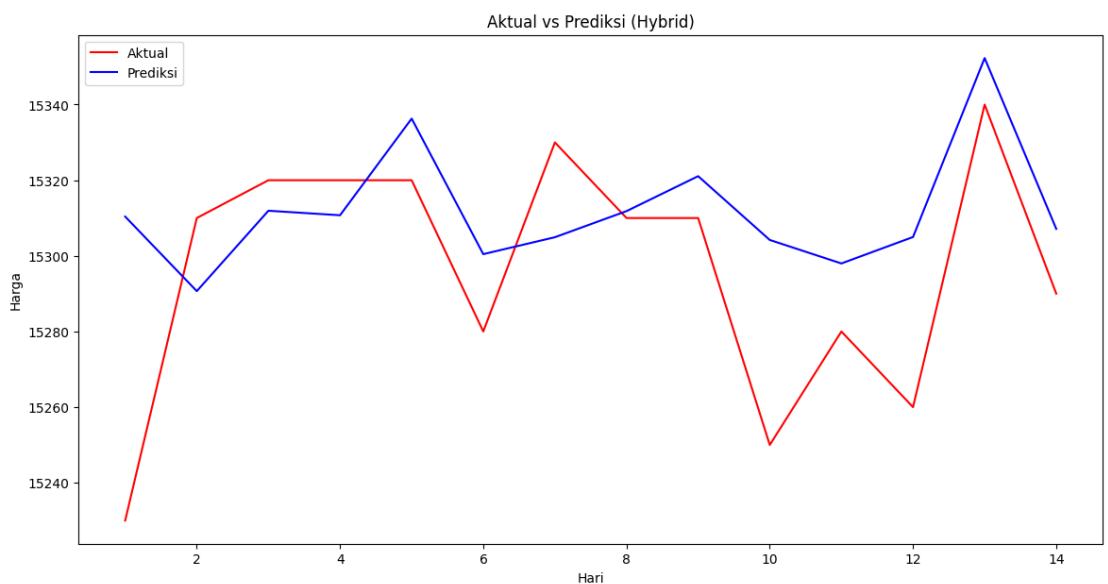
Berdasarkan tabel 4.6.3.1 diatas, model *Hybrid* dengan menggunakan rasio data latih dan validasi 80:20 pada pemodelan LSTM memiliki nilai MAE dan MAPE

yang terkecil dibanding dua model *Hybrid* lainnya. Maka model *Hybrid* SARIMA-LSTM yang akan digunakan adalah model pertama dengan *training set* 80%. Visualisasi pembagian data untuk model LSTM ditampilkan pada gambar 4.6.3.1.

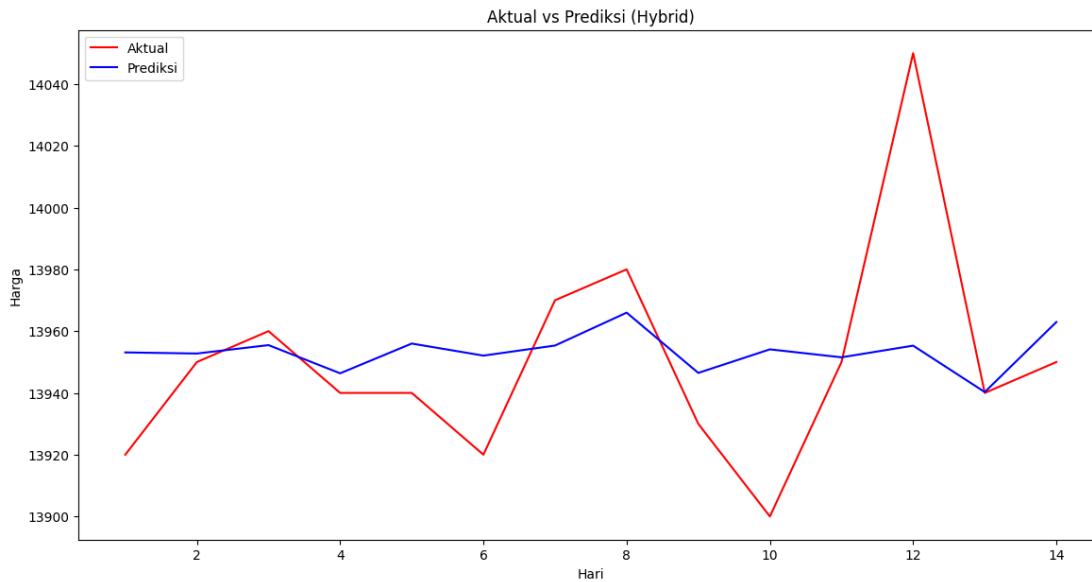


Gambar 4.6.3.1 Visualisasi pembagian data untuk LSTM pada model *Hybrid*

Plot perbandingan hasil prediksi ketiga model dan harga aktual ditampilkan pada lampiran 1 dan 2. Grafik perbandingan harga aktual dan harga prediksi untuk model *hybrid* pertama ditampilkan pada gambar 4.6.3.1 dan 4.6.3.2.



Gambar 4.6.3.2 Grafik perbandingan aktual vs prediksi harga BP (*Hybrid*)



Gambar 4.6.3.3 Grafik perbandingan aktual vs prediksi harga BM (*Hybrid*)

Pada gambar 4.6.3.2, terlihat bahwa model mampu menangkap hampir keseluruhan fluktuasi yang ada pada harga aktual beras premium. Pada hari pertama, hasil prediksi menunjukkan perbedaan yang cukup besar dibanding data aktual, namun pada hari berikutnya akurasi prediksi membaik cukup signifikan. Pada hari ke-10 dan ke-12 hasil prediksi tampaknya tidak bisa mengikuti penurunan tajam pada data aktual, namun setelahnya hasil prediksi kembali membaik sampai hari ke-14.

Berdasarkan gambar 4.6.3.3, hasil prediksi model kedua untuk beras medium menunjukkan hasil yang cukup baik, dimana perbedaan antara titik prediksi dan data aktual terlihat cukup kecil pada hampir semua prediksi. Pada hari ke-10 dan ke-12 hasil prediksi tampaknya mengalami penurunan akurasi, dimana model tidak mampu menangkap fluktuasi tajam data aktual pada hari tersebut. Setelah itu, hasil prediksi kembali membaik sampai hari terakhir. Hasil prediksi dengan model *Hybrid SARIMA-LSTM* dapat dilihat dengan jelas pada tabel 4.6.3.2 berikut.

Tabel 4.6.3.2 Hasil prediksi harga beras dengan model *Hybrid*

Tanggal	Beras Premium		Beras Medium	
	Aktual	Prediksi	Aktual	Prediksi
2024-05-18	15230	15310,38	13920	13953,11
2024-05-19	15310	15290,67	13950	13952,75
2024-05-20	15320	15311,92	13960	13955,47
2024-05-21	15320	15310,73	13940	13946,35

Tanggal	Beras Premium		Beras Medium	
	Aktual	Prediksi	Aktual	Prediksi
2024-05-22	15320	15336,29	13940	13955,98
2024-05-23	15280	15300,45	13920	13952,07
2024-05-24	15330	15304,92	13970	13955,33
2024-05-25	15310	15311,82	13980	13965,98
2024-05-26	15310	15321,06	13930	13946,47
2024-05-27	15250	15304,20	13900	13954,10
2024-05-28	15280	15297,98	13950	13951,52
2024-05-29	15260	15304,98	14050	13955,30
2024-05-30	15340	15352,29	13940	13940,28
2024-05-31	15290	15307,12	13950	13962,95

Berikut ini adalah contoh perhitungan hasil prediksi model hybrid untuk beras premium pada tanggal 22 Mei 2024.

$$\text{Prediksi SARIMA} = 15304,568$$

$$\text{Prediksi LSTM} = 31,723$$

$$\begin{aligned}\text{Prediksi Hybrid} &= \text{Prediksi SARIMA} + \text{Prediksi LSTM} \\ &= 15304,568 + 31,723 \\ &= 15336,29\end{aligned}$$

Tabel 4.6.3.3 Metrik evaluasi model *Hybrid*

Jenis Beras	MAE	MAPE (%)
Beras Premium	24,168	0,158
Beras Medium	21,680	0,155

Berdasarkan tabel 4.6.3.3, kedua model *Hybrid* menunjukkan tingkat *error* yang sangat kecil dengan nilai MAPE dibawah 1%, yaitu 0,158% untuk beras premium dan 0,155% pada beras medium. Maka dapat disimpulkan model *Hybrid* memiliki kemampuan prediksi yang sangat baik.

4.7 Evaluasi

Setelah dilakukan pengujian pada ketiga model, yaitu SARIMA, LSTM, dan *Hybrid* SARIMA-LSTM, didapatkan hasil perhitungan metrik evaluasi MAE dan MAPE yang ditampilkan pada tabel 4.7.1.

Tabel 4.7.1 Perbandingan nilai *error* seluruh model

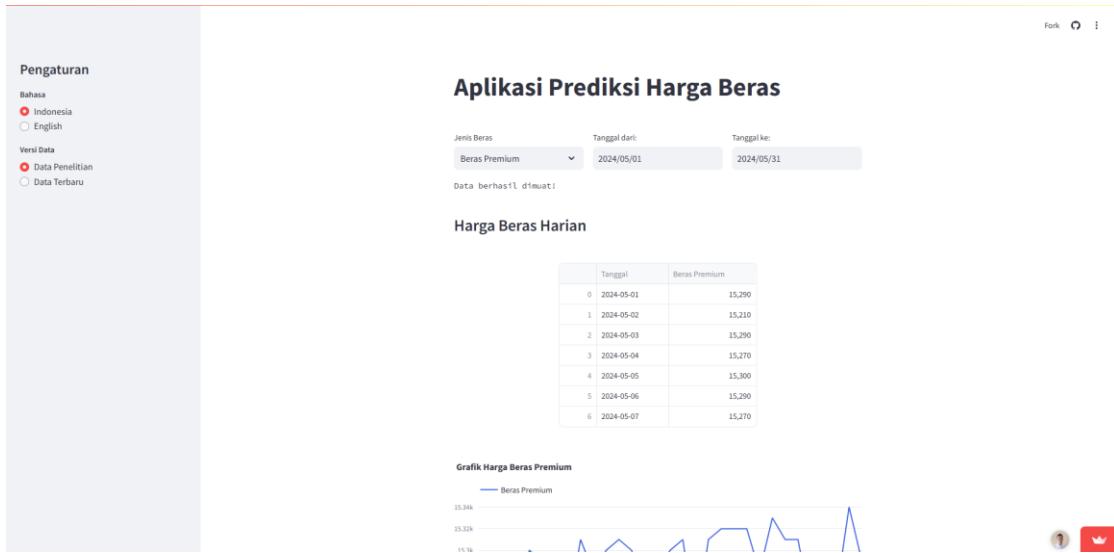
Model	MAE	MAPE (%)
SARIMA (BP)	25,296	0,165
SARIMA (BM)	22,438	0,160
LSTM (BP)	25,513	0,166
LSTM (BM)	24,090	0,172
HYBRID (BP)	24,168	0,158
HYBRID (BM)	21,680	0,155

Dari ketiga jenis model yang telah dibuat, model *Hybrid* menunjukkan nilai MAE dan MAPE terendah. Hal ini mengindikasikan model *Hybrid* SARIMA-LSTM dapat memberikan hasil yang lebih optimal dibanding dengan model individu SARIMA atau LSTM. Pada model SARIMA variabilitas prediksi lebih stabil dibandingkan dengan harga aktual, di mana prediksi berfluktuasi dalam rentang yang lebih sempit dibandingkan pada harga aktual yang mana terjadi fluktuasi yang cukup tajam hampir setiap harinya. Model LSTM memberikan hasil yang lebih baik dibanding dengan model SARIMA dalam menangkap fluktuasi pada data, namun di beberapa titik terdapat perbedaan hasil prediksi dan harga aktual tampak cukup besar yang membuat nilai *error* model ini sedikit lebih tinggi dibanding model SARIMA.

Model *Hybrid* SARIMA-LSTM memberikan hasil prediksi yang lebih optimal dari kedua model lainnya, di mana model ini dapat memberikan nilai *error* yang paling kecil dan model juga lebih baik dalam menangkap fluktuasi yang ada pada data aktual dibandingkan kedua model individu lainnya. Selanjutnya model *Hybrid* SARIMA-LSTM akan digunakan untuk di integrasikan kedalam *website*. Perbandingan Hasil prediksi seluruh model ditampilkan pada lampiran 3.

4.8 Implementasi sistem

Setelah dilakukan pelatihan model prediksi, tahap berikutnya adalah membuat sebuah *website* prediksi harga beras agar dapat digunakan oleh khalayak umum. *Website* yang dibangun menggunakan bahasa Python dan *framework* Streamlit. Setelah selesai dibuat, *source code* kemudian disimpan pada *platform* GitHub agar nantinya dapat diintegrasikan dengan akun Streamlit untuk dapat dilakukan *deploying*



Gambar 4.8.1 Tampilan awal *website* prediksi harga beras

Gambar 4.8.1 merupakan tampilan awal *website* yang telah dibangun. Pada bagian kiri terdapat menu pengaturan yang dapat digunakan pengguna untuk memilih bahasa (Indonesia dan Inggris) dan versi data. Pengguna akan diminta memilih jenis beras dan rentang tanggal yang diiginkan. Jika tidak memilih maka jenis beras yang akan terpilih adalah beras premium dan rentang tanggal adalah satu bulan terakhir pada dataset. Setelah memilih *dropdown list* tersebut akan ditampilkan daftar harga beras pada sebuah tabel dan dalam bentuk *line chart* seperti gambar 4.8.2.

Grafik Harga Beras Premium**Gambar 4.8.2 Visualisasi harga beras dengan *line chart***

Prediksi Harga 14 Hari Kedepan

Selesai!

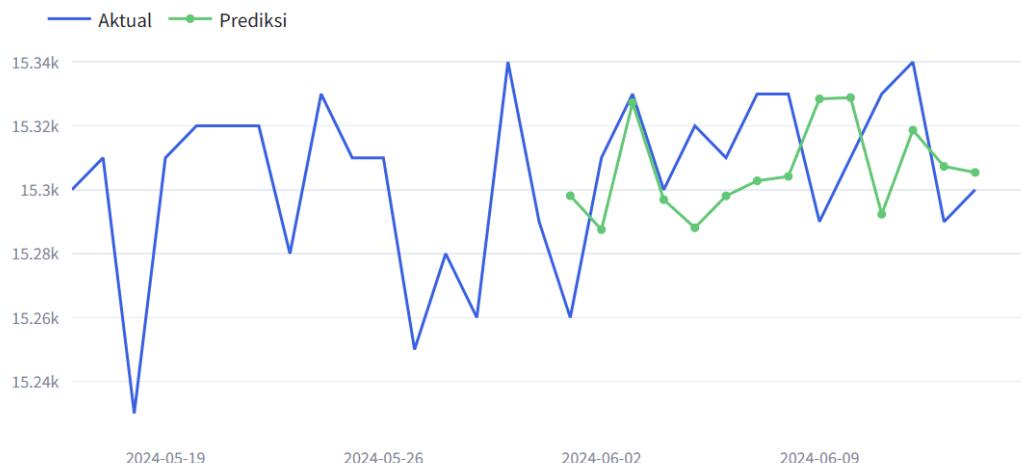
100 %

	Tanggal	Prediksi	Aktual
0	2024-06-01	15,298.1293	15,260
1	2024-06-02	15,287.5223	15,310
2	2024-06-03	15,327.2767	15,330
3	2024-06-04	15,296.9006	15,300
4	2024-06-05	15,288.0906	15,320
5	2024-06-06	15,298.078	15,310
6	2024-06-07	15,302.7803	15,330

Gambar 4.8.3 Proses prediksi harga beras pada *website*

Selanjutnya *website* akan melakukan prediksi selama 14 hari kedepan, yaitu dimulai dari tanggal 1 Juni 2024 sampai dengan 14 Juni 2024. Proses prediksi akan di visualisasi dalam *progress bar* dan setelah prediksi selesai hasilnya akan disajikan dalam sebuah tabel seperti yang terlihat pada gambar 4.8.3.

Grafik Prediksi Harga Beras Premium



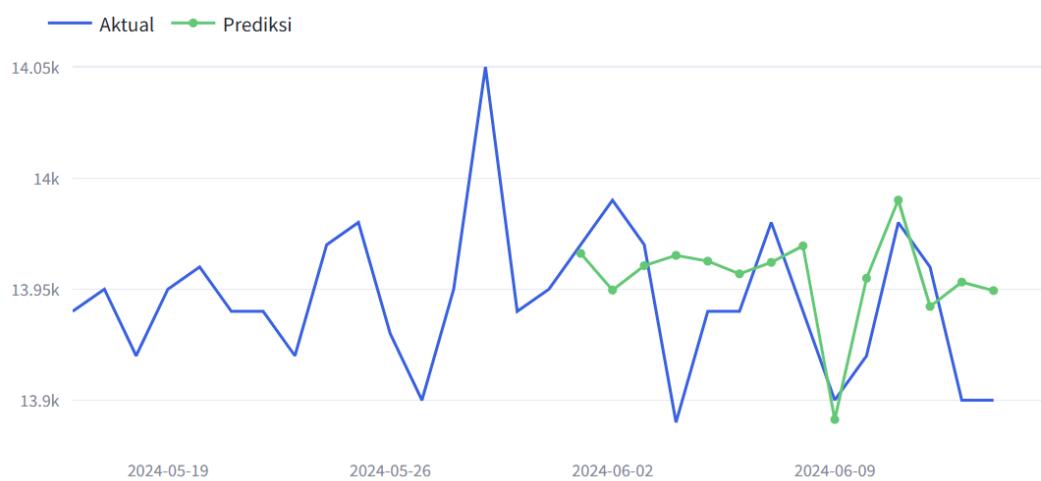
Error Prediksi

Mean Absolute Error : 21.593

Mean Absolute Percentage Error : 0.141%

Gambar 4.8.4 Grafik hasil prediksi *out of sample* harga BP 14 hari kedepan

Grafik Prediksi Harga Beras Medium



Error Prediksi

Mean Absolute Error : 27.851

Mean Absolute Percentage Error : 0.2%

Gambar 4.8.5 Grafik hasil prediksi *out of sample* harga BM 14 hari kedepan

Selain tabel, hasil prediksi akan ditampilkan dalam bentuk visualisasi *line chart* dengan penambahan titik pada prediksi untuk memperjelas posisi prediksi masing-masing hari. Garis biru menggambarkan harga aktual dan garis hijau menggambarkan hasil prediksi. Dari gambar 4.8.4, harga beras premium diprediksi paling rendah akan paling rendah pada tanggal 2 Juni dan paling tinggi pada tanggal 10 Juni. Berdasarkan gambar 4.8.5 harga beras medium diprediksi akan paling rendah pada tanggal 9 Juni dan paling tinggi pada tanggal 11 Juni. Hasil prediksi kedua model untuk kedua jenis beras dapat lebih jelas dilihat pada lampiran 3. Dikarenakan data aktual harga beras dari tanggal 1 Juni 2024 sampai 14 Juni 2024 sudah tersedia maka dapat dilakukan perhitungan nilai *error prediksi*. Nilai *error* prediksi kedua jenis beras dapat dilihat di bawah grafik *line chart* prediksi.

Pada bagian pengaturan Versi Data, pengguna dapat memilih dua jenis dataset, yaitu data yang dipakai untuk penelitian atau data terbaru hingga 31 Desember 2024. Hasil prediksi untuk tanggal 1 Januari 2024 sampai dengan 14 Januari 2024 dapat lebih jelas dilihat pada lampiran 4. Untuk *source code* pengembangan *website* akan ditampilkan pada lampiran 5. *Website* ini dapat diakses dengan URL <https://rice-price-forecasting-app.streamlit.app/>.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, penulis menyimpulkan hal-hal berikut.

1. Model SARIMA cenderung kurang akurat dalam memprediksi pola non-linear yang ada pada data. Hasil prediksi model ini cenderung lebih stabil atau konstan dibandingkan dengan harga aktual yang menunjukkan fluktuasi yang lebih besar.
2. Model LSTM menunjukkan hasil yang lebih baik, di mana hasil prediksi cenderung lebih baik dalam menangkap fluktuasi harga beras dibanding model SARIMA, namun masih terdapat *error* yang sangat tinggi pada beberapa prediksi.
3. Model *Hybrid* SARIMA-LSTM menunjukkan hasil yang paling optimal dibandingkan model individu SARIMA dan LSTM. Pada pengujian prediksi untuk 14 hari kedepan, model *Hybrid* memiliki nilai *error* yang paling kecil yaitu nilai MAE 24,168 dan MAPE 0,158% pada beras premium dan MAE 21,680 dan MAPE 0,155% pada beras medium. Sedangkan model SARIMA memiliki nilai MAE adalah 25,296 dan MAPE 0,165% pada beras premium dan MAE 22,438 dan MAPE 0,160% pada beras medium. Model LSTM menunjukkan nilai MAE 25,513 dan MAPE 0,166% pada beras premium dan MAE 24,090 dan MAPE 0,172% pada beras medium.

5.2 Saran

Beberapa rekomendasi/saran yang dapat dipertimbangkan untuk pengembangan lebih lanjut dari penelitian ini adalah sebagai berikut.

1. Penelitian selanjutnya diharapkan dapat membandingkan beberapa model atau kombinasi model lain, seperti FB Prophet, SVM, GRU, dan lain sebagainya.
2. Mempertimbangkan menggunakan strategi *multi-step ahead prediction* yang lain yang kemungkinan menghasilkan kinerja yang lebih baik dan kecepatan pelatihan yang lebih singkat.
3. Mempertimbangkan penambahan faktor eksternal, seperti jumlah stok beras dan jumlah impor dan ekspor beras.
4. Penelitian ini hanya di khususkan untuk memprediksi harga beras. Peneliti selanjutnya dapat mempertimbangkan untuk melakukan prediksi pada harga pangan lain seperti daging sapi atau cabai merah.
5. Sistem prediksi harga beras hanya dibuat dalam bentuk *website*, untuk itu dapat dipertimbangkan untuk menerapkannya ke *platform* lain seperti Android atau IOS.

DAFTAR PUSTAKA

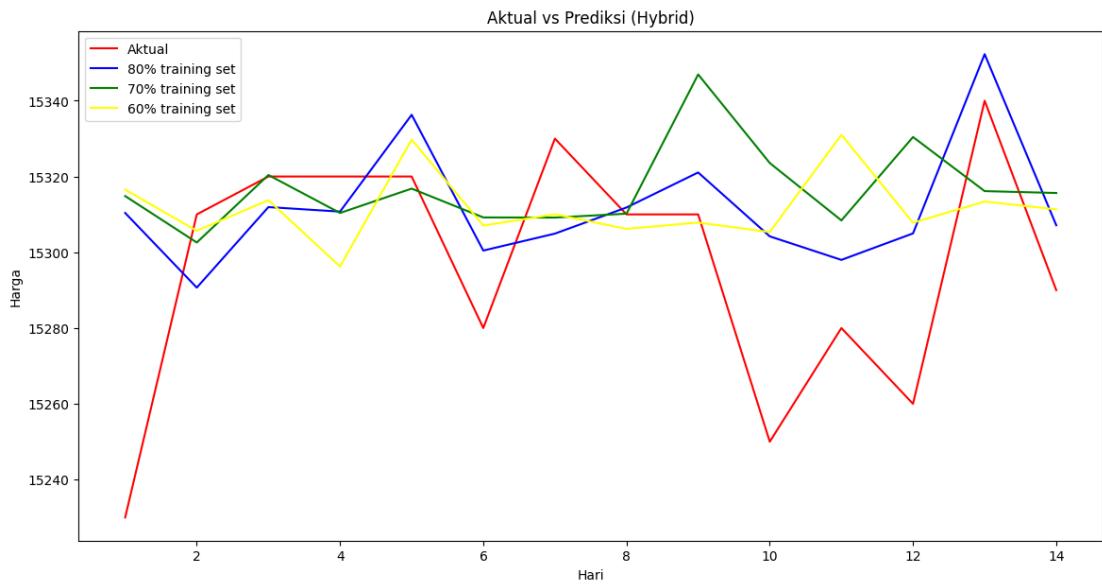
- Anwar, M. C. (2021, 03 18). *Beda Beras Premium dan Medium: Definisi dan Cara Tahu Ciri-cirinya.* Retrieved from kompas.com: <https://money.kompas.com/read/2021/03/18/164408926/beda-beras-premium-dan-medium-definisi-dan-cara-tahu-ciri-cirinya?page=all#:~:text=Beras%20premium%20maksimal%20hanya%20boleh,kuning%2Frusak%2C%20dan%20kapur>.
- Ardesfira, G., Zedha, H. F., Fazana, I., Rahmadhiyanti, J., Rahima, S., & Anwar, S. (2022). PERAMALAN NILAI TUKAR RUPIAH TERHADAP DOLLAR AMERIKA DENGAN MENGGUNAKAN METODE AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA). *JAMBURA Journal Of Probability and Statistics*, 3(2), 71-84.
- As'ad, M., Wibowo, S. S., & Sophia, E. (2017). PERAMALAN JUMLAH MAHASISWA BARU DENGAN MODEL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA). *Jurnal Informatika Merdeka Pasuruan*, 2(3), 20-33.
- Azwina, R., & Syahbudi, M. (2023). Pengaruh Fluktuasi Harga Komoditas Pangan Terhadap Inflasi di Provinsi Sumatera Utara tahun (2019-2021). *Jurnal Kajian Ekonomi & Bisnis Islam*, 238-249.
- Barrett, C. B. (1997). Heteroskedastic Price Forecasting for Food Security Management in Developing Countries. *Oxford Development Studies*, 25(2), 225-236.
- Chang, P.-C., Wang, Y.-W., & Liu, C.-H. (2007). The development of a weighted evolving fuzzy neural network for PCB sales forecasting. *Expert Systems with Applications*, 32(1), 86-96.
- Chen, K.-Y., & Wang, C.-H. (2007). A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan. *Expert Systems with Applications*, 32(1), 264-264.
- Deviana, S., Nusyirwan, Azis, D., & Ferdias, P. (2021). Analisis Model Autoregressive Integrated Moving Average Data Deret Waktu Dengan Metode Momen Sebagai Estimasi Parameter. *Jurnal Siger Matematika*, 2(2), 57-67.
- Fadil Indra Sanjaya, D. H. (2020). Prediksi Rerata Harga Beras Tingkat Grosir Indonesia dengan Long Short Term Memory. *Jurnal Teknik Informatika dan Sistem Informasi*, 7(2), 163-174.
- Hariadi, W., & Sulantari. (2021). Application of ARIMA Model for Forecasting Additional Positive Cases of Covid-19 in Jember Regency. *Enthusiastic International Journal of Statistics and Data Science*, 20-27.
- Hu, D., Zhang, Q., Ding, Y., & Zhang, D. (2022). Application of a hybrid ARIMA-LSTM model based on the SPEI for drought forecasting. *Environmental Science and Pollution Research*, 29, 4128-4144.

- Kang, J., Wang, H., Yuan, F., Wang, Z., Huang, J., & Qiu, T. (2020). Prediction of Precipitation Based on Recurrent Neural Networks in Jingdezhen, Jiangxi Province, China. *Atmosphere*, 11(3), 1-17.
- Lastinawati, E., Mulyana, A., Zahri, I., & Sriati, S. (2019). Model ARIMA untuk Peramalan Harga Beras di Kabupaten Ogan Komering Ilir Provinsi Sumatera Selatan. *Prosiding Seminar Nasional Lahan Suboptimal 2019*, 192-200.
- Makridakis, S. G., Wheelwright, S. C., & McGee, V. E. (1999). *Metode dan aplikasi peramalan* (2nd ed.). Jakarta: Erlangga.
- Masum, S., Liu, Y., & Chiverton, J. (2018). Multi-step Time Series Forecasting of Electric Load Using Machine Learning Models. *Artificial Intelligence and Soft Computing*, 10841, 148-159.
- Maulid, R. (2023, July 6). *Mengenal Streamlit, Tools Favorit Data Scientist*. Retrieved from <https://dqlab.id/>: <https://dqlab.id/mengenal-streamlit-tools-favorit-data-scientist>
- Muhamad, N. (2023, Juli 13). *Konsumsi Beras Indonesia Terbanyak Keempat di Dunia pada 2022/2023*. Retrieved from <https://databoks.katadata.co.id/>: <https://databoks.katadata.co.id/datapublish/2023/07/13/konsumsi-beras-indonesia-terbanyak-keempat-di-dunia-pada-20222023>
- Nurfadilah, K., C, F. R., & Kasse, I. (2018). PERAMALAN TINGKAT SUKU BUNGA PASAR UANG ANTAR BANK (PUAB) DENGAN VECTOR AUTOREGRESSIVE EXOGENOUS (VARX). *MSA*, 6(1), 51-60.
- Ohyver, M., & Pudjihastuti, H. (2018). Arima Model for Forecasting the Price of Medium Quality Rice to Anticipate Price Fluctuations. *Procedia Computer Science*, 707-711.
- Poomka, P., Pongsena, W., Kerdprasop, N., & Kerdprasop, K. (2019). SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit. *International Journal of Future Computer and Communication*, 8(1), 11-15.
- Ridla, M. A., Azise, N., & Rahman, M. (2023). PERBANDINGAN MODEL TIME SERIES FORECASTING DALAM MEMPREDIKSI JUMLAH KEDATANGAN WISATAWAN DAN PENUMPANG AIRPORT. *Jurnal Sistem Informasi dan Sistem Komputer*, 1-14.
- Rosiadi, D. (2011). *Analisis ekonometrika & runtun waktu terapan dengan R*. Yogyakarta: Andi.
- Rowan, Muflikhah, L., & Cholissodin, I. (2022). Peramalan Kasus Positif COVID-19 di Jawa Timur menggunakan Metode Hybrid ARIMA-LSTM. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4146-4153.
- Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1394-1401.
- Sirisha, U. M., Belavagi, M. C., & Attigeri, G. (2022). Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison. *IEEE Access*, 10, 124715-124727.

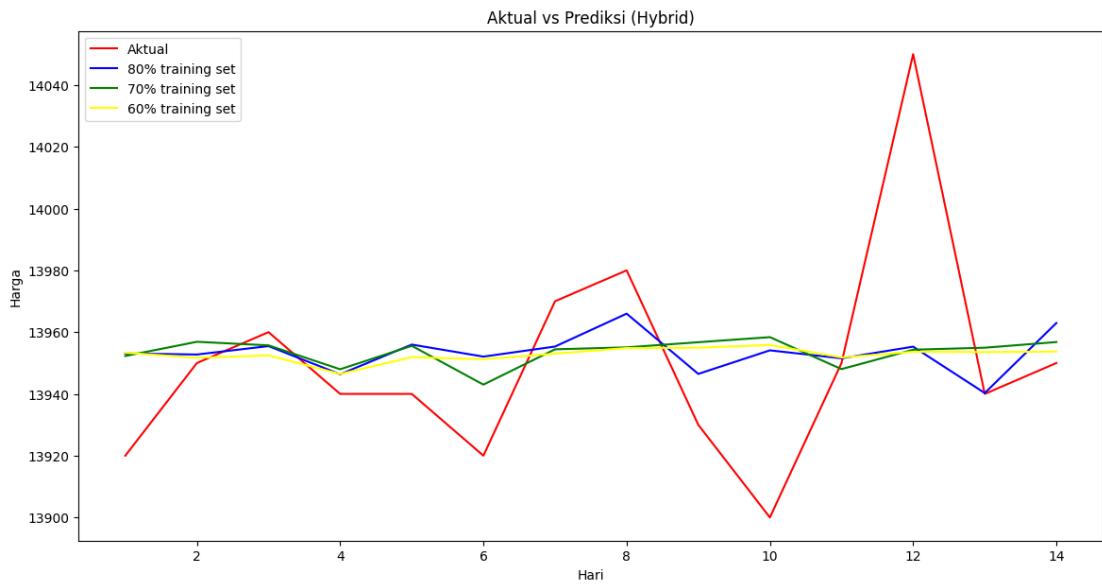
- Sitorus, V. B., Wahyuningsih, S., & Hayati, M. N. (2017). Peramalan dengan Metode Seasonal Autoregressive Integrated Moving Average (SARIMA) di Bidang Ekonomi (Studi Kasus: Inflasi Indonesia). *Jurnal EKSPONENSIAL*, 8(1), 17-26.
- Snowflake Inc. (n.d.). *Welcome to Streamlit Community Cloud*. Retrieved from <https://docs.streamlit.io/>: <https://docs.streamlit.io/deploy/streamlit-community-cloud>
- Visakha, M., & Wustqa, D. U. (2023). PERAMALAN HARGA BERAS MENGGUNAKAN METODE HYBRID AUTOREGRESSIVE INTEGRATED MOVING AVERAGE DAN NEURAL NETWORK (ARIMA-NN). *JURNAL KAJIAN DAN TERAPAN MATEMATIKA*, 148-162.
- Wei, W. W. (2006). *Time Series Analysis: Univariate & Multivariate Methods. Second Edition*. Boston: Pearson-Addison Wesley.
- Widi, H. (2024, March 1). *Beras Picu Inflasi Lagi, Harga Nasi Lauk Naik Cukup Tinggi*. Retrieved from kompas.id: <https://www.kompas.id/baca/ekonomi/2024/03/01/beras-picu-inflasi-lagi-harga-nasi-lauk-naik-cukup-tinggi>
- Wiranda, L., & Sadikin, M. (2019). Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk PT. Mestika Farma. *Jurnal Nasional Pendidikan Teknik Informatika*, 184-196.
- Wulandari, R. A., & Gernowo, R. (2019). METODE AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA) DAN METODE ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) DALAM ANALISIS CURAH HUJAN. *BERKALA FISIKA*, 41-48.
- Yati, E., Devianto, D., & Asdi, Y. (2013). TRANSFORMASI BOX-COX PADA ANALISIS REGRESI LINIER SEDERHANA. *Jurnal Matematika UNAND*, 2(2), 115-122.
- Yu, L., Feng, T., Li, T., & Cheng, L. (2022). Demand Prediction and Optimal Allocation of Shared Bikes Around Urban Rail Transit Stations. *Urban Rail Transit*, 9(1), 57–71.
- Zhao, L., Li, Z., & Qu, L. (2022). Forecasting of Beijing PM2.5 with a *hybrid* ARIMA model based on integrated AIC and improved GS fixed-order methods and seasonal decomposition. *Heliyon*, 1-16.

LAMPIRAN

Lampiran 1 Plot perbandingan ketiga model *hybrid* (Beras Premium)



Lampiran 2 Plot perbandingan ketiga model *hybrid* (Beras Medium)



Lampiran 3 Perbandingan hasil prediksi seluruh model

Tanggal	Aktual		SARIMA		LSTM		Hybrid	
	(BP)	(BM)	(BP)	(BM)	(BP)	(BM)	(BP)	(BM)
2024-05-18	15230	13920	15307,63	13947,59	15277,61	13942,34	15310,38	13953,11
2024-05-19	15310	13950	15302,75	13949,42	15300,37	13934,49	15290,67	13952,75
2024-05-20	15320	13960	15308,90	13948,83	15298,66	13945,22	15311,92	13955,47
2024-05-21	15320	13940	15305,05	13941,39	15303,50	13938,91	15310,73	13946,35
2024-05-22	15320	13940	15304,56	13947,39	15304,50	13946,34	15336,29	13955,98
2024-05-23	15280	13920	15302,91	13947,50	15289,64	13931,60	15300,45	13952,07
2024-05-24	15330	13970	15307,41	13947,41	15317,10	13930,31	15304,92	13955,33
2024-05-25	15310	13980	15305,77	13949,38	15294,81	13968,72	15311,82	13965,98
2024-05-26	15310	13930	15307,97	13950,40	15309,97	13949,76	15321,06	13946,47
2024-05-27	15250	13900	15308,88	13950,07	15304,56	13958,08	15304,20	13954,10
2024-05-28	15280	13950	15305,42	13945,94	15330,87	13952,53	15297,98	13951,52
2024-05-29	15260	14050	15302,98	13949,27	15315,20	13952,45	15304,98	13955,30
2024-05-30	15340	13940	15308,39	13949,33	15322,79	13957,42	15352,29	13940,28
2024-05-31	15290	13950	15307,16	13949,28	15321,00	13969,29	15307,12	13962,95

Lampiran 4 Hasil Prediksi 1 Juni 2024 – 14 Juni 2024

Tanggal	Beras Premium	Beras Medium
2024-06-01	15298,12	13966,07
2024-06-02	15287,52	13949,56
2024-06-03	15327,27	13960,56

Tanggal	Beras Premium	Beras Medium
2024-06-04	15296,90	13965,18
2024-06-05	15288,09	13962,59
2024-06-06	15298,07	13956,88
2024-06-07	15302,78	13962,04
2024-06-08	15304,16	13969,48
2024-06-09	15328,44	13891,31
2024-06-10	15328,82	13954,88
2024-06-11	15292,32	13990,09
2024-06-12	15318,67	13942,19
2024-06-13	15307,29	13953,17
2024-06-14	15305,43	13949,35

Lampiran 5 Hasil Prediksi 1 Januari 2024 – 14 Januari 2024

Tanggal	Beras Premium	Beras Medium
2025-01-01	15228,26	13827,81
2025-01-02	15256,70	13815,88
2025-01-03	15279,68	13825,69
2025-01-04	15301,11	13812,05
2025-01-05	15276,73	13809,53
2025-01-06	15285,95	13831,97
2025-01-07	15282,54	13822,37
2025-01-08	15252,18	13816,47
2025-01-09	15296,87	13847,44
2025-01-10	15314,65	13825,93
2025-01-11	15285,71	13826,60
2025-01-12	15289,72	13811,59
2025-01-13	15296,25	13817,30
2025-01-14	15290,41	13830,79

Lampiran 6 Source Code Pengembangan Website

```
import streamlit as st
from datetime import date, timedelta, datetime
from plotly import graph_objs as go
import pandas as pd
import numpy as np
from tensorflow.keras.models import load_model
```

```

import joblib
import time
import math
from scipy import stats
from scipy.special import inv_boxcox
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error

language = [
    {
        'nav': ['Versi Data', 'Bahasa'],
        'sdbar_title': 'Pengaturan',
        'list_data': ['Data Penelitian', 'Data Terbaru'],
        'lang': ['Indonesia','English'],
        'title' : 'Aplikasi Prediksi Harga Beras',
        'sl_box' : ['Jenis Beras', 'Tanggal dari:', 'Tanggal ke:'],
        'rice_type': ['Beras Premium', 'Beras Medium'],
        'load_state' : ['Data berhasil dimuat!', 'Data gagal dimuat!'],
        'header' : ['Harga Beras Harian','Prediksi Harga 14 Hari Kedepan', 'Error
Prediksi'],
        'table': ['Tanggal','Prediksi','Aktual'],
        'actual_chart' : ['Grafik Harga Beras Premium','Grafik Harga Beras Medium'],
        'pred_status': ['Memproses...', 'Selesai!'],
        'pred_chart' : ['Grafik Prediksi Harga Beras Premium','Grafik Prediksi Harga
Beras Medium', 'Aktual', 'Prediksi']
    },
    {
        'nav': ['Data Version', 'Language'],
        'sdbar_title': 'Settings',
        'list_data' : ['Research Data', 'Latest data'],
        'lang' : ['Indonesia', 'English'],
        'title' : 'Rice Price Prediction App',
        'sl_box' : ['Rice Type','Date from:', 'Date to:'],
        'rice_type': ['Premium Rice', 'Medium Rice'],
        'load_state' : ['Data loaded successfully!', 'Data failed to load!'],
        'header' : ['Daily Rice Price', 'Price Predictions for the Next 14 Days', 'Prediction
Error'],
        'table': ['Date','Prediction','Actual'],
        'actual_chart' : ['Premium Rice Price Chart','Medium Rice Price Chart'],
        'pred_status': ['Processing...', 'Done!'],
        'pred_chart' : ['Premium Rice Price Prediction Chart','Medium Rice Price
Prediction Chart', 'Actual', 'Prediction']
    }
]

```

```

#emojis = https://www.webfx.com/tools/emoji-cheat-sheet/
st.set_page_config(page_title = "Aplikasi Prediksi Harga Beras",
                   page_icon = "📈")
lang = None
with st.sidebar:
    if "lang_choose" not in st.session_state:
        st.session_state.lang_choose = "Indonesia"
    for i in range(len(language[0]['lang'])):
        if st.session_state.lang_choose == language[i]['lang'][i]:
            st.session_state.label = f"**{language[i]['nav'][1]}**"
            st.session_state.index = i
            st.session_state.title = language[i]['sdbar_title']
            lang = i
            continue
    st.title(st.session_state.title)
    if "label" not in st.session_state:
        st.session_state.label = f"**{language[0]['nav'][1]}**"
    if "index" not in st.session_state:
        st.session_state.index = 0

    st.radio(
        st.session_state.label,
        language[0]['lang'],
        key="lang_choose",
        index=st.session_state.index,
    )

    st.radio(
        f"**{language[lang]['nav'][0]}**",
        language[lang]['list_data'],
        key="data_choose",
        index=0,
    )
data_idx = language[lang]['list_data'].index(st.session_state.data_choose)
st.title(f" {language[lang]['title']} ")
def add_space(n_space):
    for i in range(n_space):
        st.text(" ")
add_space(2)

if (data_idx == 0):
    df_beras = pd.read_excel("datasets/export-eceran.xlsx", sheet_name = 1)
elif (data_idx == 1):

```

```

df_beras = pd.read_excel("datasets/export-eceran-baru.xlsx", sheet_name = 1)
df_beras = df_beras.rename(columns={' Komoditas (Rp) ': 'tanggal', 'Beras Premium' : 'beras_premium', 'Beras Medium': 'beras_medium'})

def data_prep(data):
    data['tanggal'] = pd.to_datetime(data['tanggal'], dayfirst=True)
    data[data.columns[1]] = data[data.columns[1]].replace('-', np.nan)
    data.set_index('tanggal', inplace=True)
    data = data.asfreq('D')
    data[data.columns[0]] = data[data.columns[0]].interpolate(method='linear')
    data = data.astype({data.columns[0]: int})
    return data

data_idx = language[lang]['list_data'].index(st.session_state.data_choose)
if(data_idx == 0):
    beras_premium = data_prep(df_beras[['tanggal', 'beras_premium']].iloc[:-14].copy())
    beras_premium1 = data_prep(df_beras[['tanggal', 'beras_premium']].copy())
    beras_medium = data_prep(df_beras[['tanggal', 'beras_medium']].iloc[:-14].copy())
    beras_medium1 = data_prep(df_beras[['tanggal', 'beras_medium']].copy())
elif(data_idx == 1):
    beras_premium = data_prep(df_beras[['tanggal', 'beras_premium']].copy())
    beras_medium = data_prep(df_beras[['tanggal', 'beras_medium']].copy())

first_date = beras_premium.iloc[0].name
last_date = beras_premium.iloc[-1].name
rice_type = language[lang]['rice_type']
col1, col2, col3 = st.columns(3)
default_date = last_date
with col1 :
    selected_rice = st.selectbox(language[lang]['sl_box'][0], rice_type)
with col2 :
    start_date = st.date_input(language[lang]['sl_box'][1], value =
    default_date.replace(day = 1),
                            min_value = first_date,
                            max_value=last_date)
with col3:
    end_date = st.date_input(language[lang]['sl_box'][2], value = default_date,
                            min_value = first_date,
                            max_value = last_date)

def load_datesets(ticker, date1, date2):

```

```

rice_index = language[lang]['rice_type'].index(ticker)
if rice_index == 0:
    final_data = beras_premium.copy()
    data = beras_premium.loc[date1 : date2].copy()
    final_data['beras_premium'], lmd =
stats.boxcox(beras_premium['beras_premium'])
    rice_type = 'premium'
elif rice_index == 1:
    final_data = beras_medium.copy()
    data = beras_medium.loc[date1 : date2].copy()
    final_data['beras_medium'], lmd = stats.boxcox(beras_medium['beras_medium'])
    rice_type = 'medium'

return data, final_data, ticker, rice_type, lmd, rice_index

data, final_data, name, rice_type, lmd, rice_index = load_datesets(selected_rice,
start_date, end_date)
data.index = data.index.strftime('%Y-%m-%d')
data = data.reset_index()
data = data.rename(columns={'tanggal':language[lang]['table'][0], data.columns[1] :
name})
load_data_state = st.text("")

if(data is not None):
    load_data_state.text(language[lang]['load_state'][0])
else:
    load_data_state.text(language[lang]['load_state'][1])
add_space(1)

st.subheader(language[lang]['header'][0])
add_space(2)
col1, col2, col3 = st.columns([1,2,1])
with col2:
    rows = 8
    st.dataframe(data, height = rows * 35 + 3, width = 400)

fig = go.Figure()
fig.add_trace(go.Scatter(x = data.iloc[:,0],
y = data.iloc[:,1],
name = data.columns[1],
line=dict(color='#2B60DE')))
fig.layout.update(title_text = language[lang]['actual_chart'][rice_index],
xaxis=dict(tickformat='%Y-%m-%d'),

```

```

showlegend = True,
legend=dict(orientation="h",
            yanchor="bottom",
            y = 1,
            xanchor="center",
            x = 0.1,
            font = dict(size = 14)))
st.plotly_chart(fig)\

add_space(1)
st.subheader(language[lang]['header'][1])
add_space(1)
window_size = 5
steps = 14

def load_model_final(data_ver):
    lstm_model = []
    if (data_ver == 0):
        path = f'models/{rice_type}/model_final_forecasting/'
    elif (data_ver == 1):
        path = f'models/{rice_type}/model_final_forecasting_baru/'
        sc_path = path+"scaler_final_hybrid.pkl"
        sarima_path = path+"sarima_model_final_hybrid.pkl"
        sc = joblib.load(sc_path)
        sarima_model = joblib.load(sarima_path)
    for i in range(steps):
        model_path = path+f'lstm_model_final_{i}_hybrid.keras'
        model = load_model(model_path)
        lstm_model.append(model)

    return sc, sarima_model, lstm_model

def make_data_direct(data, window_size, n_steps):
    x1, y1 = [], []
    for i in range(len(data) - window_size - n_steps + 1):
        x1.append(data[i:(i + window_size)])
        y1.append(data[(i + window_size):(i + window_size + n_steps)])
    return np.array(x1), np.array(y1)

def direct_lstm_pred(models, x, scaler, n_steps):
    # predict
    direct_pred = np.zeros((x.shape[0], n_steps))

```

```

pred_status = st.text(language[lang]['pred_status'][0])
bar = st.progress(0)
progress_status = st.empty()
bar_step = math.floor(100/steps)
j_temp = 1
all_progress = 0
for i, regs in enumerate(models):
    n_progress = bar_step
    if i == (steps-1):
        n_progress += 2
    direct_pred[:, i] = regs.predict(x).flatten()
    all_progress += n_progress
    for j in range(j_temp, all_progress + 1):
        bar.progress(j)
        progress_status.write(str(j) + " %")
    j_temp = all_progress
pred_status.text(language[lang]['pred_status'][1])
direct_pred = scaler.inverse_transform(direct_pred)

return direct_pred

def residualForLstm(actual, pred, scaller):
    temp = actual.copy()
    temp['pred'] = pred
    temp['residual'] = temp.iloc[:, 0] - temp.iloc[:, 1]
    residual = pd.DataFrame(temp['residual'][1:], columns=['residual'])
    resid_scaled = scaller.fit_transform(residual)

    return resid_scaled

def make_data_input_only(data, window_size):
    x_inp = []
    for i in range(len(data) - window_size + 1):
        x_inp.append(data[i:(i + window_size)])

    return np.array(x_inp)

def hybrid_model_predict_final(sm_model, lm_model, X, scaller):
    sm_pred= sm_model.predict(start = X.index[0], end = X.index[-1])
    sm_forecast= sm_model.forecast(steps=14)
    sm_forecast = inv_boxcox(sm_forecast, lmd)
    data_temp = inv_boxcox(X, lmd)
    data_temp['pred'] = inv_boxcox(sm_pred, lmd)

```

```

data_temp['residual'] = data_temp.iloc[:, 0] - data_temp.iloc[:, 1]
residual = pd.DataFrame(data_temp['residual'][1:], columns=['residual'])
resid_sc = scaller.fit_transform(residual)
x1 = make_data_input_only(resid_sc, window_size)
direct_pred = direct_lstm_pred(lm_model, x1, scaller, steps)
direct_pred1 = direct_pred[direct_pred.shape[0]-1:]
direct_pred1 = direct_pred1.transpose()
all_pred = sm_forecast.to_frame()
all_pred['resid_pred'] = direct_pred1
all_pred['result'] = all_pred.iloc[:,0] + all_pred.iloc[:,1]

return all_pred['result'].to_frame()

sc, sarima_model, lstm_model = load_model_final(data_idx)
pred = hybrid_model_predict_final(sarima_model, lstm_model, final_data, sc)
final_pred = pred
last_date = final_data.index.max()
new_date = pd.date_range(start = last_date + timedelta(days = 1), periods = steps)
df_pred = pd.DataFrame({f'{final_data.columns[0]}': final_pred.iloc[:,0].values},
index = new_date)
if (data_idx == 0):
    if (rice_type == 'premium'):
        df_pred[language[lang]['table'][2]] = beras_premium1.iloc[-14:]
    elif (rice_type == 'medium'):
        df_pred[language[lang]['table'][2]] = beras_medium1.iloc[-14:]
else:
    df_pred[language[lang]['table'][2]] = np.nan
add_space(1)
col1, col2, col3 = st.columns([1,2,1])
with col2:
    rows = 15
    df_pred2 = df_pred.copy()
    df_pred2.index = df_pred2.index.strftime('%Y-%m-%d')
    df_pred2 = df_pred2.reset_index()
    df_pred2 = df_pred2.rename(columns={'index' : language[lang]['table'][0],
df_pred2.columns[1] : language[lang]['table'][1]})
    st.dataframe(df_pred2, height = rows * 35 + 3, width = 400)
add_space(1)
fig = go.Figure()
subset_data = final_data.copy()
subset_data[subset_data.columns[0]] =
inv_boxcox(subset_data[subset_data.columns[0]], lmd)
if (data_idx == 0):

```

```

if (rice_type == 'premium'):
    subset_data = beras_premium1
elif (rice_type == 'medium'):
    subset_data = beras_medium1
subset_data = subset_data[-30:]
fig.add_trace(go.Scatter(x = subset_data.index,
                          y = subset_data.iloc[:,0],
                          name = language[lang]['pred_chart'][2],
                          line=dict(color='#2B60DE')))
fig.add_trace(go.Scatter(x = df_pred.index,
                          y = df_pred.iloc[:,0],
                          name = language[lang]['pred_chart'][3],
                          line=dict(color='#50C878')))
fig.layout.update(title_text = language[lang]['pred_chart'][rice_index],
                  xaxis=dict(
                      tickformat='%Y-%m-%d',
                  ),
                  showlegend = True,
                  legend=dict(orientation="h",
                             yanchor="bottom",
                             y = 1,
                             xanchor="center",
                             x = 0.1,
                             font = dict(size = 14)))
st.plotly_chart(fig)

def pred_error(actual, pred):
    mae = mean_absolute_error(actual, pred)
    mape = mean_absolute_percentage_error(actual, pred) * 100
    return mae, mape

if (data_idx == 0):
    mae, mape = pred_error(df_pred2.iloc[:,2], df_pred2.iloc[:,1])
    mae = mae.round(3)
    mape = mape.round(3)
else :
    mae, mape = ' ', ' '
st.subheader(language[lang]['header'][2])
st.write(f"**Mean Absolute Error** : {mae}")
st.write(f"**Mean Absolute Percentage Error** : {mape}%")

```