

BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Penelitian yang berkaitan dengan pengembangan sistem prediksi harga beras berbasis web menggunakan metode Long Short-Term Memory (LSTM) telah banyak dilakukan oleh berbagai peneliti di bidang kecerdasan buatan dan analisis deret waktu.

Penelitian oleh (Hidayat, 2024) berjudul “Rice Price Prediction with Long Short-Term Memory (LSTM) Neural Network” membahas penerapan model LSTM untuk memprediksi harga beras di Indonesia menggunakan data harga harian dan variabel cuaca. Hasil penelitian menunjukkan bahwa LSTM mampu memberikan hasil prediksi yang stabil dengan tingkat kesalahan rendah (RMSE = 0,054), menunjukkan efektivitasnya dalam menangani pola fluktuatif harga beras.

Selanjutnya, penelitian oleh (Devi Dwi Purwanto, 2024) berjudul “Comparison of Premium Rice Price Prediction in East Java with ARIMA and LSTM” melakukan perbandingan dua model, yaitu ARIMA dan LSTM, terhadap data harga beras premium dari Bapanas selama periode Maret 2021–Oktober 2024. Hasil penelitian membuktikan bahwa model LSTM menghasilkan akurasi lebih tinggi dibanding ARIMA karena mampu mempelajari hubungan non-linear dalam data historis harga beras.

Penelitian ketiga oleh (Sunoto & Siahaan, 2025) berjudul “Analysis of Rice Yield Prediction with MLPRegressor and Long Short-Term Memory Models” menggunakan dua model pembelajaran mesin, yaitu MLPRegressor dan LSTM, untuk memprediksi produktivitas beras di 34 provinsi di Indonesia. Berdasarkan

hasil penelitian, model LSTM memperoleh akurasi sebesar 94,12% dan menunjukkan kemampuan terbaik dalam mengenali pola deret waktu pada sektor pertanian.

Dari ketiga penelitian tersebut dapat disimpulkan bahwa metode LSTM memiliki kemampuan yang unggul dalam melakukan prediksi berbasis deret waktu. Oleh karena itu, penelitian ini mengadopsi metode LSTM dan mengintegrasikannya dalam sebuah aplikasi web interaktif menggunakan Streamlit untuk memudahkan pengguna dalam mengakses hasil prediksi harga beras secara langsung.

2.2 Pandas dan NumPy

Pandas dan NumPy merupakan dua pustaka utama dalam bahasa pemrograman Python yang digunakan untuk pengolahan data. Pandas berfungsi dalam manipulasi data tabular seperti membaca, menulis, memfilter, dan mengelompokkan data dalam bentuk DataFrame. Sedangkan NumPy digunakan untuk operasi matematis dan komputasi numerik, seperti perhitungan statistik, matriks, serta operasi vektor.

Dalam penelitian ini, Pandas digunakan untuk membaca data harga beras dari file CSV dan melakukan pembersihan serta konversi tipe data tanggal. NumPy berperan dalam proses normalisasi dan penyusunan data input menjadi bentuk array tiga dimensi sebelum dimasukkan ke dalam model LSTM.

2.3 Scikit-Learn (MinMaxScaler dan Train-Test Split)

Scikit-Learn adalah pustaka Python yang digunakan untuk data preprocessing dan pemodelan dasar pembelajaran mesin. Pada penelitian ini, MinMaxScaler digunakan untuk menormalkan data harga beras agar semua nilai

berada dalam rentang 0 hingga 1, sehingga model LSTM dapat belajar dengan stabil tanpa dipengaruhi oleh skala besar pada data.

Selain itu, fungsi `train_test_split` digunakan untuk membagi data menjadi dua bagian, yaitu training set dan testing set, agar performa model dapat diuji secara objektif. Pembagian data ini membantu menghindari overfitting pada model.

2.4 Keras dan Metode Long Short-Term Memory (LSTM)

Keras merupakan pustaka tingkat tinggi untuk membangun dan melatih model deep learning di atas backend TensorFlow. Salah satu arsitektur jaringan saraf yang didukung oleh Keras adalah LSTM (Long Short-Term Memory).

LSTM merupakan jenis Recurrent Neural Network (RNN) yang mampu mengingat informasi jangka panjang. Arsitektur LSTM terdiri dari tiga gerbang utama, yaitu input gate, forget gate, dan output gate. Kombinasi ketiga gerbang ini memungkinkan LSTM mempertahankan informasi penting dari data sebelumnya dan membuang informasi yang tidak relevan. Pada penelitian ini, LSTM digunakan untuk mengenali pola historis harga beras dan menghasilkan prediksi harga untuk periode berikutnya secara akurat.

2.5 Streamlit dan Plotly

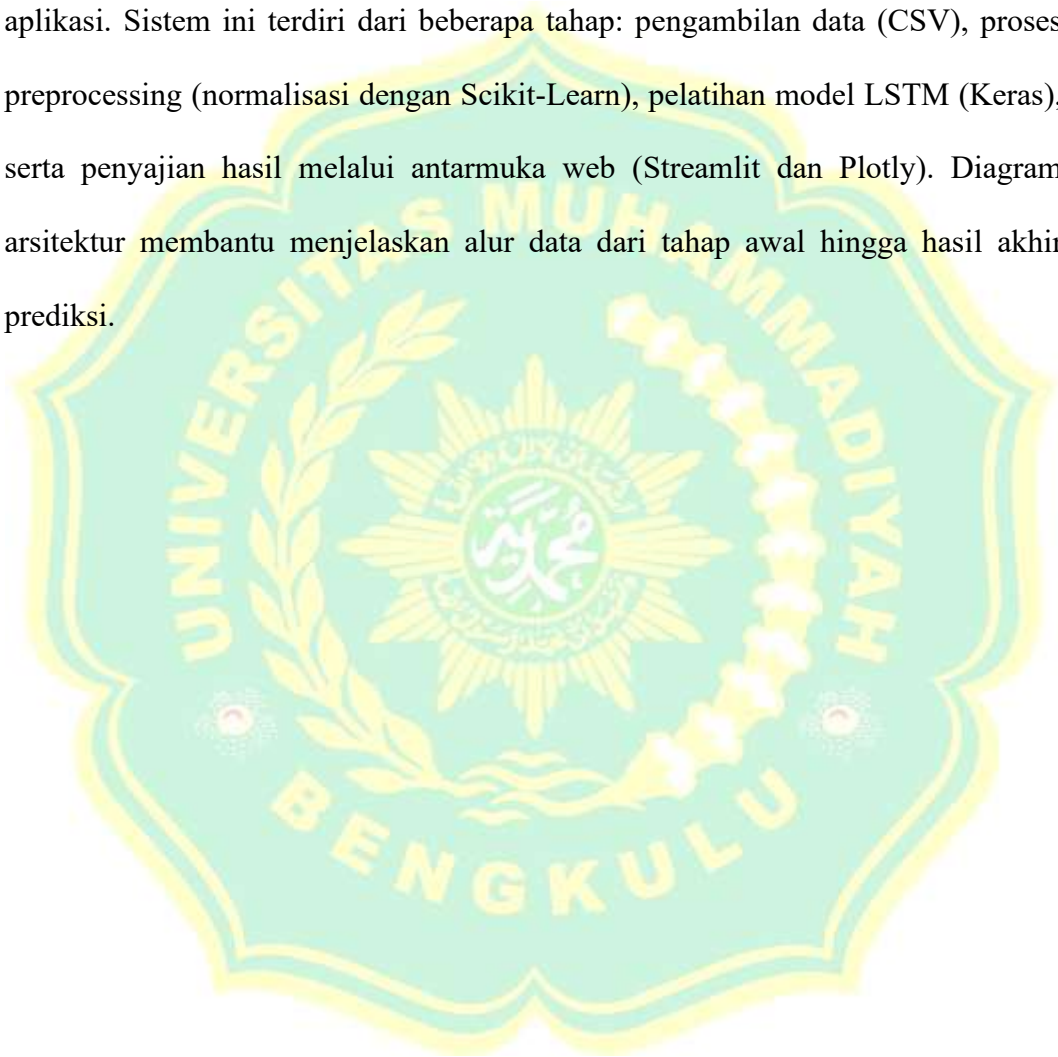
Streamlit merupakan framework Python yang memudahkan pengembangan aplikasi berbasis web untuk keperluan analisis data dan machine learning. Dengan Streamlit, pengguna dapat membuat antarmuka interaktif hanya dengan beberapa baris kode tanpa memerlukan pengembangan web konvensional.

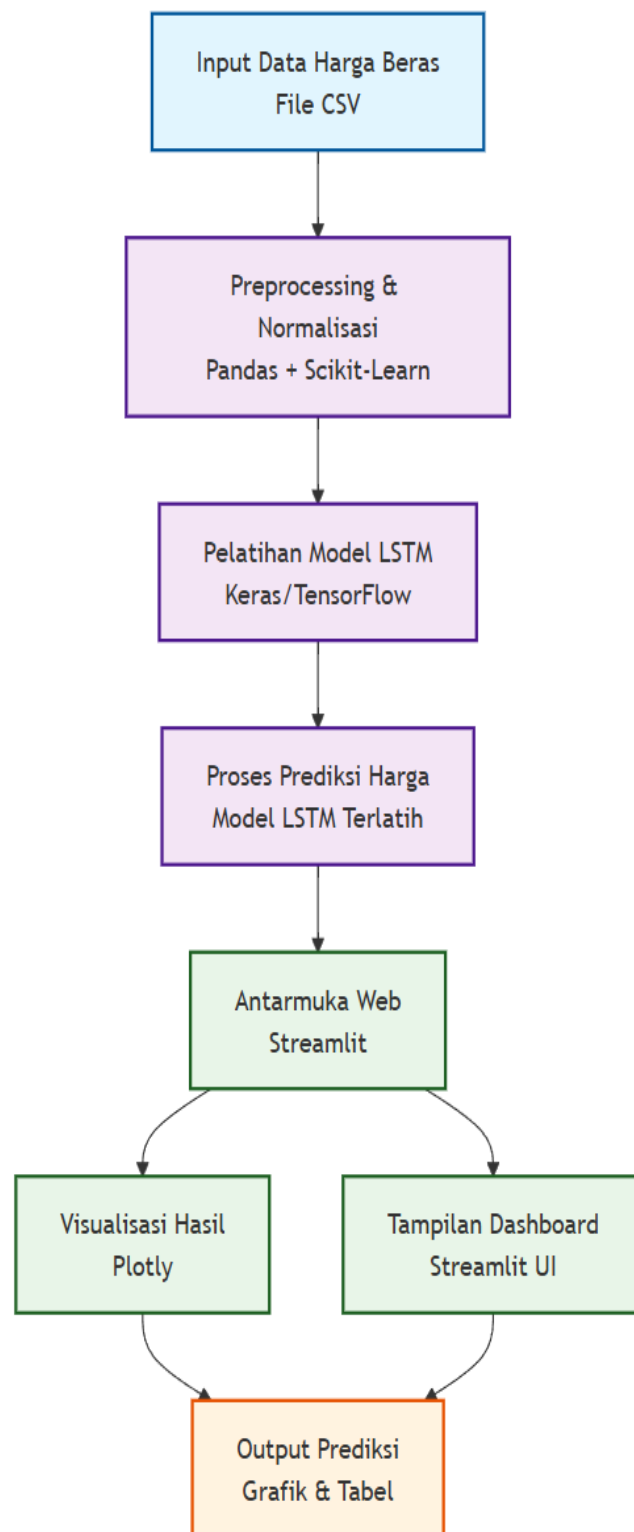
Plotly digunakan untuk menghasilkan visualisasi data yang dinamis dan interaktif, seperti grafik tren harga dan hasil prediksi model. Dalam penelitian ini,

kombinasi Streamlit dan Plotly digunakan untuk menampilkan data historis harga beras, hasil pelatihan model, serta grafik prediksi harga beras pada periode mendatang.

2.6 Diagram Arsitektur Sistem

Arsitektur sistem menggambarkan hubungan antar-komponen utama pada aplikasi. Sistem ini terdiri dari beberapa tahap: pengambilan data (CSV), proses preprocessing (normalisasi dengan Scikit-Learn), pelatihan model LSTM (Keras), serta penyajian hasil melalui antarmuka web (Streamlit dan Plotly). Diagram arsitektur membantu menjelaskan alur data dari tahap awal hingga hasil akhir prediksi.

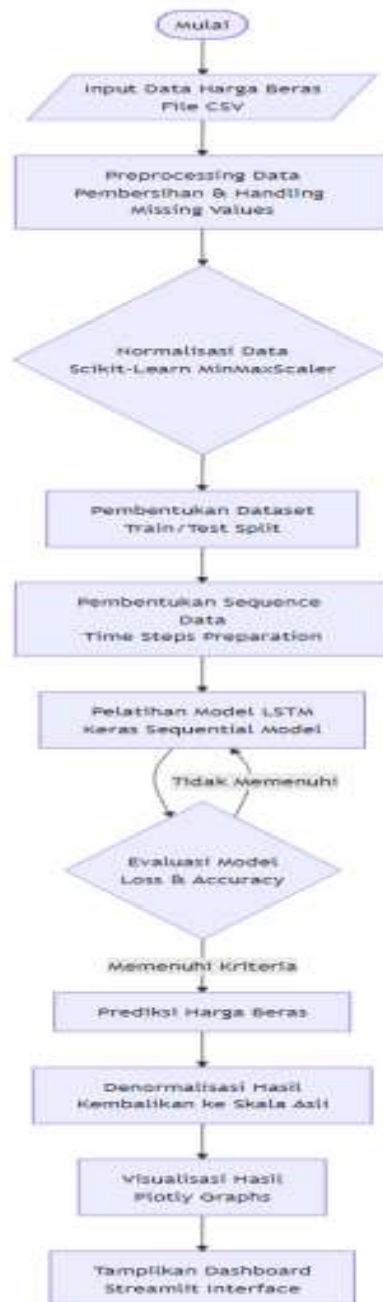




Gambar 1 Diagram Arsitektur Sistem Prediksi Harga Beras

2.7 Flowchart Sistem

Flowchart sistem menggambarkan alur kerja logis dari keseluruhan proses aplikasi. Dimulai dari input data harga beras, proses normalisasi, pelatihan model LSTM, prediksi harga, hingga visualisasi hasil. Flowchart ini berfungsi sebagai panduan pengembangan sistem agar setiap tahapan berjalan sesuai prosedur.



Gambar 2 Flow Chart Sistem Prediksi Harga Beras

2.8 Website dan Deployment

Aplikasi prediksi ini dibangun menggunakan teknologi web agar dapat diakses secara luas melalui peramban (browser). Dengan menggunakan Streamlit, proses deployment dapat dilakukan dengan cepat ke server lokal maupun cloud. Teknologi web memungkinkan hasil prediksi ditampilkan secara real-time dengan antarmuka yang sederhana dan informatif bagi pengguna.

2.9 Penyimpanan dan Manajemen Data

Data harga beras disimpan dalam format CSV sebagai media penyimpanan utama. Format ini dipilih karena sederhana, mudah diakses, dan kompatibel dengan pustaka Python seperti Pandas dan NumPy. Penyimpanan data yang baik diperlukan agar proses analisis dan pelatihan model dapat dilakukan secara efisien tanpa kehilangan integritas data.