

Penerapan Metode *Long Short Term Memory* untuk Memprediksi Harga Beras di Indonesia

Anggun Kurniatul Hidayah¹, Ricky Eka Putra²

^{1,2} Teknik Informatika, Universitas Negeri Surabaya

¹anggun.20006@mhs.unesa.ac.id

²rickyeka@unesa.ac.id

Abstrak— Beras merupakan makanan pokok masyarakat di Indonesia. Berdasarkan data Sistem Pemantauan Pasar dan Kebutuhan Pokok (SP2KP) Kementerian Perdagangan, harga beras di Indonesia terus meningkat sejak Agustus 2022 [1], hal ini berdampak pada tingkat kemiskinan, tingkat inflasi, dan stabilitas ekonomi. Oleh karena itu, pemerintah membutuhkan Solusi agar dapat menghindari dampak dari naik turunnya harga beras di Indonesia. Metode *Long Short Term Memory (LSTM)* memiliki kemampuan dalam menangani masalah ketergantungan jarak jauh pada data berurutan, sehingga cocok digunakan untuk memprediksi harga beras. Kombinasi parameter yang digunakan adalah metode normalisasi data, pembagian data, *layer*, *batch size*, *epoch*, dan *learning rate*. Hasil evaluasi model menggunakan *MAPE (Mean Absolute Percentage Error)* menunjukkan bahwa *LSTM* dapat memprediksi harga beras dengan akurasi 98.57% dan nilai *MAPE* 1.43%. Hasil prediksi terbaik didapatkan dengan menggunakan parameter metode normalisasi data *StandardScaler*, pembagian data validasi 10%, data latih 80%, data uji 10%, *layer* 2, *batch size* 4, *epoch* 40, dan *learning rate* 0.01, dengan pembagian data secara acak. Performa metode *LSTM* terbukti memiliki performa yang lebih baik. Untuk mengetahui performa *LSTM*, dilakukan perbandingan dengan metode *ARIMA (Autoregressive Integrated Moving Average)* dan *RNN (Recurrent Neural Network)*. Nilai *MAPE* terbaik dari *LSTM* 1.43%, *ARIMA* 10.41%, dan *RNN* 3.53%. Adapun hasil akurasi terbaik *LSTM* ialah 98.57%, *ARIMA* 89.59%, dan *RNN* 96.47%.

Kata Kunci— *Long Short Term Memory*, *Autoregressive Integrated Moving Average*, *Recurrent Neural Network*, *Mean Absolute Percentage Error*, prediksi, beras

I. PENDAHULUAN

Makanan pokok bagi sebagian besar populasi adalah beras. Sebagian populasi di Asia menjadikan beras sebagai makanan pokok sehari-hari, salah satunya adalah Indonesia. Hingga awal tahun 2018, harga beras Indonesia cenderung lebih tinggi dibandingkan rata-rata harga beras internasional, menurut Organisasi Pangan dan Pertanian (FAO). Bahkan di Filipina, yang mempunyai status perdagangan sama dengan Indonesia, harga beras relatif rendah, yaitu beras Filipina dihargai \$0,82 per kilogram dan beras Indonesia dihargai \$1,01 per kilogram [2]. Berdasarkan data *Food and Agriculture Organization (FAO)*, trend produksi beras global cenderung meningkat dalam 10 tahun terakhir. Pada periode 2021-2022 tercatat bahwa produksi beras mencapai hingga 520,8 juta ton [1].

Beras sebagai kebutuhan makanan pokok masyarakat Indonesia yang dikonsumsi setiap hari, membuat masyarakat Indonesia tetap membeli dan mengonsumsi beras meskipun

harga beras dalam keadaan naik maupun turun. Berdasarkan data Sistem Pemantauan Pasar dan Kebutuhan Pokok (SP2KP) Kementerian Perdagangan, harga beras di Indonesia terus meningkat sejak Agustus 2022 [1]. Hal ini menjadi perhatian pemerintah dan para pemerhati untuk mencari cara untuk mengontrol harga beras di Indonesia. Naik dan turunnya harga beras dapat berdampak pada tingkat kemiskinan, tingkat inflasi, dan stabilitas ekonomi [1]. Perubahan harga beras menjadi faktor penting yang dapat mempengaruhi kehidupan masyarakat yang memiliki pendapatan terbatas. Untuk mencegah dan mengurangi dampak dari naik turunnya harga beras, maka pemerintah perlu melakukan pemantauan dan mengontrol harga beras yang naik turun. Untuk itu dibutuhkan informasi yang dapat memprediksi harga beras dari tahun ke tahun di Indonesia.

Metode *LSTM (Long Short Term Memory)* merupakan pengembangan lebih lanjut dari arsitektur *RNN (Recurrent Neural Network)* yang pertama kali diperkenalkan oleh Hochreiter & Schmidhuber pada tahun 1997 [3]. Metode ini dirancang untuk menangani masalah ketergantungan jarak jauh dalam data berurutan, seperti teks, suara, dan deret waktu. Metode *RNN* tidak dapat belajar mengkorelasikan informasi karena memori lama yang disimpan di *RNN* akan ditimpa atau diganti dengan memori baru seiring waktu. Di sisi lain, *LSTM* tidak memiliki kelemahan tersebut. *LSTM* dapat mengatur memori untuk setiap input menggunakan sel memori dan unit gerbang [4].

LSTM memiliki tiga gerbang kendali: gerbang masukan, gerbang keluaran, dan gerbang lupa. Gerbang masukan digunakan untuk mengontrol aliran aktivasi masukan ke dalam sel memori. Gerbang keluaran digunakan untuk mengontrol aliran keluaran aktivasi sel di seluruh jaringan. Forget Gate menskalakan status yang ada dalam sel sebelum menambahkannya sebagai input ke sel melalui koneksi sel berulang, sehingga memungkinkannya mengatur ulang memori sel secara adaptif. Dengan kata lain, ketiga gerbang kendali dirancang untuk memperbarui, menyimpan, dan melupakan [5].

Penelitian tentang prediksi atau peramalan harga beras sebelumnya telah banyak dilakukan. Berdasarkan penelitian yang dilakukan oleh Anandyani dkk (2021) mengenai prediksi harga beras grosir menggunakan metode *ARIMA (Autoregressive Integrated Moving Average)*, memiliki hasil prediksi yang cukup akurat. Metode *ARIMA* ialah metode yang biasa digunakan untuk melakukan peramalan deret waktu (*time series*) [6]. Metode *ARIMA* merupakan generalisasi dari *Autoregressive Moving Average (ARMA)*

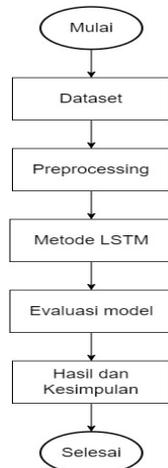
yang dapat dimanfaatkan untuk peramalan atau prediksi nilai dalam deret waktu (*time series*). Metode *ARIMA* memungkinkan Anda membuat prakiraan jangka pendek dengan hasil yang akurat. Oleh karena itu metode *ARIMA* banyak digunakan untuk peramalan atau prediksi suatu nilai [7].

Kemampuan *LSTM* dalam menangani masalah ketergantungan jarak jauh pada data berurutan, membuat metode *LSTM* sangat cocok digunakan untuk memprediksi harga beras [6]. Dengan mengumpulkan data harga beras dari waktu ke waktu, maka data tersebut dapat diolah dengan *LSTM* untuk mendapatkan prediksi harga beras yang akurat. Pada penelitian sebelumnya metode ini juga sering digunakan untuk memprediksi harga emas, minyak mentah, saham, dan masih banyak lagi.

Berdasarkan permasalahan di atas, penelitian ini bertujuan untuk menerapkan metode *long short-term memory (LSTM)* dalam prediksi harga beras di Indonesia. Dengan memanfaatkan dataset yang disediakan secara publik oleh Badan Pusat Statistik, penulis akan mencari parameter terbaik untuk mendapatkan hasil prediksi yang akurat menggunakan metode *LSTM*. Selain itu, untuk mengetahui performa dari metode *LSTM*, akan dilakukan perbandingan hasil prediksi menggunakan metode *ARIMA* dan *RNN*.

II. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah metode *Long Short Term Memory (LSTM)*. Alur penelitian adalah rencana atau tahapan yang akan diambil pada suatu penelitian. Adapun pembuatan alur penelitian bertujuan untuk memberikan gambaran kerangka kerja yang terstruktur sehingga peneliti dapat menjalankan penelitian secara efisien. Gbr. 1 merupakan gambaran alur penelitian yang digunakan dalam penelitian ini.



Gbr. 1 Alur Penelitian

A. Dataset

Dataset yang digunakan dalam penelitian ini adalah *dataset* “Harga Rata-Rata Beras Tingkat Grosir Indonesia” yang diunduh dari platform resmi BPS (Badan Pusat Statistik) yaitu “bps.go.id”. *Dataset* ini berisi data rata-rata harga beras di tingkat grosir di Indonesia pada tahun 2010 hingga 2024.

Dataset ini selalu melakukan pembaruan data di setiap akhir bulan. Saat penelitian ini dilakukan, total data harga beras dalam *dataset* ini adalah 170 data yang dimulai pada Januari 2010 sampai dengan Februari 2024.

B. Preprocessing

Preprocessing adalah tahapan yang dilakukan untuk mempersiapkan data sebelum berlanjut ke tahap pemodelan. Pada penelitian ini, terdapat tiga tahap *preprocessing* yang dilakukan. Adapun tahapan *preprocessing* tersebut yaitu:

1) Seleksi dan Transformasi Data

Berdasarkan buku yang ditulis oleh Achmad Sudiro (2022), berdasarkan pendapat para ahli, berikut ini ialah beberapa definisi terkait seleksi:

- Menurut Yoder (1981), seleksi merupakan proses dimana calon pekerja dibagi menjadi dua, yakni calon pekerja yang diterima dan ditolak.
- Menurut Hasibuan, seleksi adalah kegiatan menyeleksi dan menentukan calon pegawai yang boleh atau tidak boleh bekerja pada suatu organisasi. Setiap perusahaan juga memiliki persyaratan khusus mengenai jumlah tenaga kerja yang dibutuhkan. [8].

Dalam konteks penelitian ini, calon pekerja atau calon pegawai yang dimaksud dalam definisi tersebut adalah data. *Transformasi data* adalah proses pengolahan data mentah. Data mentah hasil penyaringan akan diolah dan diubah sesuai kaidah yang berlaku [9]. Dalam penelitian ini seleksi dan transformasi data ialah memilah data apa saja yang akan digunakan dan data apa yang akan dibersihkan dan mengubah format data agar dapat diolah dengan baik dengan menghapus data atau kolom yang tidak dibutuhkan, mengambil data harga dan tanggal, dan mengubah format tanggal menjadi yyyy-mm-dd.

2) Normalisasi Data

Normalisasi data pada penelitian ini menggunakan metode *MinMaxScaler* dan *StandardScaler*. *MinMaxScaler* adalah metode normalisasi data yang bertujuan untuk mentransformasikan nilai-nilai data ke dalam rentang antara 0 dan 1. *MinMaxScaler* digunakan untuk membuat distribusi data menjadi seragam. Selain itu *MinMaxScaler* juga digunakan untuk mempertahankan hubungan yang relatif antar nilai [10]. *StandardScaler* dikenal juga sebagai *Z-Score Normalization* atau *standardization*. *StandardScaler* adalah salah satu teknik normalisasi data yang bertujuan untuk mentransformasikan nilai-nilai data sebagai proses awal pemodelan. Penggunaan *StandardScaler* dilakukan dengan pemanggilan melalui library *Scikitlearn*. Dalam proses *StandardScaler* untuk dapat menggeser distribusi, dilakukan proses standarisasi fitur dengan cara mengurangi mean, untuk kemudian dibagi dengan standar deviasi [11].

3) Pembagian Data

Split Data atau pembagian data adalah metode membagi data dalam kumpulan data. Fungsi data split adalah membagi *dataset* menjadi dua bagian data: data latih dan data uji, dengan proporsi tertentu. Data latih digunakan untuk melatih model untuk melakukan prediksi pada machine learning. Data uji diperlukan untuk melakukan evaluasi model.

Dengan adanya split data atau pembagian data, akan memudahkan dalam melakukan prediksi pada suatu penelitian [12]. Pada penelitian ini membagi dataset menjadi tiga data yang terdiri dari data validasi, data latih, dan data uji. Pembagian data dibagi menjadi dua, yaitu 15% data validasi, 70% data latih, dan 15% data uji, dan 10% data validasi, 80% data latih, dan 10% data uji.

C. Metode LSTM

Metode *Long Short Term Memory* atau *LSTM* adalah metode *deep learning* yang merupakan evolusi dari arsitektur *RNN (Recurrent Neural Network)* [3]. Penelitian ini menggunakan metode *LSTM* karena metode *LSTM* dapat menangani masalah ketergantungan jarak jauh pada data berurutan. Mengatur memori pada setiap masukannya dapat dilakukan *LSTM* dengan menggunakan *gate units* dan *memory cells*. Sehingga memori yang lama tidak akan tergantikan atau tertimpa dengan memori yang baru. Sebelum data diolah lebih lanjut, diperlukan perancangan model parameter untuk membangun model prediksi. Parameter yang digunakan dalam penelitian ini adalah metode *normalisasi data*, pembagian data, total *layer*, *batch size*, *epoch*, *learning rate*, dan *optimizer*.

Tabel I merupakan kombinasi parameter metode normalisasi. Parameter metode normalisasi data yang digunakan pada penelitian ini ada dua, yaitu *MinMaxScaler* dan *StandardScaler*.

TABEL I

KOMBINASI PARAMETER METODE NORMALISASI

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	<i>MinMaxScaler</i>	80% : 10% : 10%	2	4	40
2.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	40

Tabel II merupakan kombinasi dari parameter pembagian data. Parameter pembagian data yang digunakan ada dua cara, yaitu 10% data validasi, 80% data latih, dan 10% data uji, dan 15% data validasi, 70% data latih, dan 15% data uji.

TABEL II

KOMBINASI PARAMETER PEMBAGIAN DATA

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	40
2.	<i>StandardScaler</i>	70% : 15% : 15%	2	4	40

Tabel III merupakan kombinasi parameter *layer* dengan total *layer* yang digunakan dalam penelitian terdapat lima total *layer* yaitu *layer 2*, *layer 3*, *layer 4*, *layer 5*, dan *layer 6*.

TABEL III

KOMBINASI PARAMETER LAYER

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	40
2.	<i>StandardScaler</i>	80% : 10% : 10%	3	4	40
3.	<i>StandardScaler</i>	80% : 10% : 10%	4	4	40
4.	<i>StandardScaler</i>	80% : 10% : 10%	5	4	40
5.	<i>StandardScaler</i>	80% : 10% : 10%	6	4	40

Tabel IV merupakan kombinasi dari parameter *batch size*, dimana parameter *batch size* yang digunakan dalam penelitian ini ada lima yaitu 4, 8, 16, 32, dan 64.

TABEL IV

KOMBINASI PARAMETER BATCH SIZE

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	40
2.	<i>StandardScaler</i>	80% : 10% : 10%	2	8	40
3.	<i>StandardScaler</i>	80% : 10% : 10%	2	16	40
4.	<i>StandardScaler</i>	80% : 10% : 10%	2	32	40
5.	<i>StandardScaler</i>	80% : 10% : 10%	2	64	40

Pada tabel V terdapat kombinasi parameter *epoch*. Parameter *epoch* yang digunakan pada penelitian ini ada lima nilai *epoch* yaitu *epoch 10*, *epoch 20*, *epoch 30*, *epoch 40*, dan *epoch 50*.

TABEL V

KOMBINASI PARAMETER EPOCH

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	10
2.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	20
3.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	30
4.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	40
5.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	50

Setelah model parametrik selesai dibuat, model tersebut diproses menggunakan metode *Long Short Term Memory (LSTM)*. Pada metode *LSTM* terdapat empat proses fungsi aktivasi untuk setiap masukan ke neuron terdekat yang disebut

unit gerbang. Satuan gerbang terdiri dari gerbang lupa, gerbang masukan, gerbang sel, dan gerbang keluaran. Saat melatih model menggunakan *LSTM*, komputasi untuk setiap masukan dimulai dengan urutan gerbang lupa, gerbang masukan, gerbang sel, dan gerbang keluaran [13].

Proses yang terjadi di *Forget-Gates* memproses dan memilih semua informasi tentang data masukan untuk menentukan data mana yang disimpan dan data mana yang tidak lagi digunakan atau dibuang di dalam sel memori. Fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid, dan keluaran dari fungsi ini antara 0 dan 1. Jika outputnya 1, semua data disimpan. Sebaliknya jika outputnya 0 maka datanya dibuang. [14].

Gerbang Masukan Ada dua gerbang yang dijalankan. Gerbang pertama menggunakan fungsi sigmoid untuk menentukan nilai mana yang akan diperbarui. Gerbang kedua menggunakan fungsi aktivasi Tanh untuk membuat vektor nilai baru dan kemudian menyimpannya dalam sel memori. Kedua gerbang tersebut ditempatkan bersamaan untuk proses pembaharuan sel memori [14].

Cell gating adalah proses dimana nilai sel memori lama diubah dengan nilai sel memori baru. Nilai ini diperoleh dengan menggabungkan nilai yang terdapat pada gerbang *forget* dan *input* [14].

Gerbang keluar memiliki dua gerbang berjalan. Gerbang pertama yang menentukan nilai keluaran dari sel memori menggunakan fungsi aktivasi sigmoid. Gerbang kedua kemudian menggunakan fungsi aktivasi Tanh untuk menempatkan nilai di sel memori. Kedua gerbang tersebut dikalikan untuk menghasilkan nilai keluaran [14].

D. Evaluasi Model

Evaluasi model yang digunakan dalam penelitian ini dilakukan dengan menggunakan hasil prediksi dari *LSTM* dan data uji, untuk kemudian dilakukan penghitungan *MAPE*. *Mean Absolute Percentage Error* atau *MAPE* adalah metode evaluasi model dan pengujian akurasi yang ditampilkan dalam bentuk persentase. *MAPE* sering digunakan dalam pengukuran, terutama pada prediksi atau peramalan. Hal ini dikarenakan *MAPE* mudah untuk dibaca karena bentuknya dalam persentase. Evaluasi model menggunakan *MAPE* dapat dilakukan dengan menemukan nilai persentase *MAPE* terendah. Semakin rendah persentasenya, maka semakin akurat hasilnya [15].

E. Hasil dan Kesimpulan

Setelah tahap evaluasi model, maka akan diketahui hasil perhitungan dari metode evaluasi, yaitu *MAPE*. Tahap terakhir pada penelitian ini adalah menentukan model prediksi yang memiliki tingkat akurasi paling tinggi. Seperti yang telah terlihat pada gambar alur, dalam penentuan model prediksi yang paling akurat adalah dengan cara membandingkan hasil prediksi dengan data yang asli atau data yang sebenarnya. Dalam hal ini hasil prediksi akan ditampilkan dalam bentuk grafik perbandingan harga. Sehingga dapat diketahui dengan jelas perbandingan perbedaan dari hasil prediksi dengan data asli. Untuk menentukan tingkat akurasi hasil prediksi

dilakukan dengan cara menemukan nilai *MAPE* terkecil sehingga didapatkan nilai persentase akurasinya. Untuk mengetahui bahwa hasil prediksi menggunakan *LSTM* memiliki hasil yang lebih akurat dibandingkan metode yang telah banyak digunakan sebelumnya, maka dalam penelitian ini akan dilakukan perbandingan hasil prediksi menggunakan metode *LSTM* dengan hasil prediksi menggunakan metode *ARIMA* (*AutoRegressive Integrated Moving Average*) yang merupakan metode statistik yang seringkali digunakan dalam analisis deret waktu dan metode *Recurrent Neural Network* (*RNN*).

III. HASIL DAN PEMBAHASAN

A. Preprocessing

1) Seleksi dan Transformasi Data

Rata-rata Harga Beras di Tingkat Perdagangan Besar (Grosir) Indonesia (Perusahaan)													
Harga Beras	2023												
	Januari	Februari	Maret	April	Mei	Juni	Juli	Agustus	September	Oktober	November	Desember	Tahunan
Harga Beras	11.647,91	11.990,12	12.041,64	12.092,38	12.102,70	12.115,81	12.141,72	12.265,68	13.036,96	13.315,29	13.380,40	13.458,06	12.465,72

Gbr. 2 Dataset Sebelum Seleksi dan Transformasi (Sumber: www.bps.go.id)

Dataset pada Gbr. 2 di atas adalah tabel dataset sebelum dilakukan proses seleksi dan transformasi data. Dataset tersebut dalam laman “bps.go.id” ditampilkan per tahun [16]. Seperti contoh pada Gbr. 2 merupakan tabel dataset harga beras di tahun 2023 yang ditampilkan harga beras dari bulan Januari hingga Desember. Dalam dataset tersebut juga terdapat data rata-rata harga beras tahunan. Untuk memudahkan dalam pemrosesan dataset, maka perlu dilakukan seleksi dan transformasi data. Dalam hal ini data yang diperlukan untuk penelitian ini adalah data tanggal dan harga beras setiap per bulan dari bulan Januari 2010 hingga Februari 2024, sedangkan untuk data rata-rata harga beras tahunan tidak diperlukan. Selanjutnya dilakukan transformasi data dengan mengubah format data tanggal menjadi yyyy-mm-dd. Dataset yang awalnya berbentuk vertikal diubah menjadi horizontal dengan dua kolom yaitu tanggal dan harga untuk memudahkan dalam pemrosesan data. Tabel VI di bawah ini merupakan tabel dataset yang telah melalui proses seleksi dan transformasi data, dimana dalam dataset ini telah diseleksi dengan menghapus data kolom yang tidak diperlukan dan telah dilakukan transformasi data.

TABEL VI

DATASET SETELAH SELEKSI DAN TRANSFORMASI

Tanggal	Harga
2010-01-31	6.702,49
2010-02-28	6.887,83
2010-03-31	6.853,78
2010-04-30	6.761,49
2010-05-31	6.772,46
2010-06-30	6873.45
2010-07-31	7025.88
2010-08-31	7317.51
2010-09-30	7350.8
2010-10-31	7391.27
2010-11-30	7457.08
2010-12-31	7617.46

2011-01-31	7853.48
...	...
2023-11-30	13.380,4
2023-12-31	13.458,06
2024-01-31	13.588
2024-02-29	14.397

2) Normalisasi Data

Dataset yang telah melalui proses seleksi dan transformasi data selanjutnya akan dilakukan normalisasi data. Metode normalisasi data yang digunakan dalam penelitian ini ada dua yaitu metode normalisasi data dengan *StandardScaler* dan *MinMaxScaler*.

```
scaler_X = StandardScaler()
scaler_y = StandardScaler()

X_train = scaler_X.fit_transform(X_train)
X_val = scaler_X.transform(X_val)
X_test = scaler_X.transform(X_test)

y_train = scaler_y.fit_transform(y_train.reshape(-1, 1)).flatten()
y_val = scaler_y.transform(y_val.reshape(-1, 1)).flatten()
y_test = scaler_y.transform(y_test.reshape(-1, 1)).flatten()
```

Gbr. 3 Normalisasi Data *StandardScaler*

```
scaler_X = MinMaxScaler()
scaler_y = MinMaxScaler()

X_train = scaler_X.fit_transform(X_train)
X_val = scaler_X.transform(X_val)
X_test = scaler_X.transform(X_test)

y_train = scaler_y.fit_transform(y_train.reshape(-1, 1)).flatten()
y_val = scaler_y.transform(y_val.reshape(-1, 1)).flatten()
y_test = scaler_y.transform(y_test.reshape(-1, 1)).flatten()
```

Gbr. 4 Normalisasi Data *MinMaxScaler*

3) Pembagian Data

Tahap preprocessing yang ketiga yaitu pembagian data. Data dibagi menjadi tiga data, yaitu data latih, data validasi, dan data uji.

```
n_train = int(0.8 * len(X))
n_val = int(0.1 * len(X))
n_test = len(X) - n_train - n_val

X_train, X_val, X_test = X[:n_train], X[n_train:n_train + n_val], X[n_train + n_val:]
y_train, y_val, y_test = y[:n_train], y[n_train:n_train + n_val], y[n_train + n_val:]
```

Gbr. 5 Pembagian Data 80% Data Latih, 10% Data Validasi, dan 10% Data Uji

```
n_train = int(0.7 * len(X))
n_val = int(0.15 * len(X))
n_test = len(X) - n_train - n_val

X_train, X_val, X_test = X[:n_train], X[n_train:n_train + n_val], X[n_train + n_val:]
y_train, y_val, y_test = y[:n_train], y[n_train:n_train + n_val], y[n_train + n_val:]
```

Gbr. 6 Pembagian Data 70% Data Latih, 15% Data Validasi, dan 15% Data Uji

B. Penerapan Metode LSTM

Prediksi harga beras menggunakan metode *LSTM* ini dilakukan dengan menggunakan bahasa pemrograman python. Prediksi dilakukan dengan memanfaatkan layanan yang disediakan oleh google, yaitu *Google Colab*. Dalam penerapannya, diperlukan parameter-parameter yang bervariasi untuk mendapatkan hasil prediksi harga beras yang akurat. Data akan melalui tahap preprocessing terlebih dahulu, untuk kemudian diolah menggunakan metode *LSTM* dengan memanfaatkan berbagai parameter yang dikombinasikan untuk menemukan hasil terbaik. Untuk mengetahui hasil

prediksi yang tepat, maka akan dilakukan evaluasi model menggunakan *MAPE* untuk mengetahui presentase kesalahan dan akurasi. Selain itu juga akan ditampilkan grafik yang menunjukkan perbandingan harga asli dengan harga prediksi.

```
k = params['k']
kf = KFold(n_splits=k, shuffle=True, random_state=42)
fold = 1
for train_index, test_index in kf.split(X):
    print(f"Training Fold {Fold}/{k}")
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    # Split data latih menjadi data latih dan validasi
    n_train = int(params['train_ratio'] * len(X_train))
    n_val = len(X_train) - n_train

    X_train, X_val = X_train[:n_train], X_train[n_train:]
    y_train, y_val = y_train[:n_train], y_train[n_train:]
    train_percentage = len(X_train) / len(X) * 100 # Persentase dari total data
    val_percentage = len(X_val) / len(X) * 100
    test_percentage = len(X_test) / len(X) * 100
    # Normalisasi data
    if params['scaler'] == 'MinMaxScaler':
        scaler_X = MinMaxScaler()
        scaler_y = MinMaxScaler()
    else:
        scaler_X = StandardScaler()
        scaler_y = StandardScaler()

    X_train = scaler_X.fit_transform(X_train)
    X_val = scaler_X.transform(X_val)
    X_test = scaler_X.transform(X_test)
```

Gbr. 7 Pembagian Data dan Normalisasi Data

Code pada gambar tersebut adalah code yang digunakan untuk membagi data latih, data validasi, data uji, dan melakukan normalisasi data. Adapun dalam penelitian ini menggunakan dua cara pembagian data yaitu 80% data latih, 10% data validasi, 10% data uji, dan 70% data latih, 15% data validasi, dan 15% data uji. Untuk metode normalisasi data juga menggunakan dua cara yaitu *MinMaxScaler* dan *StandardScaler*.

```
model = Sequential([
    LSTM(64, activation='relu', return_sequences=True, input_shape=(X_train.shape[1], 1)),
    LSTM(32, activation='relu'),
    Dense(1)
])
```

Gbr. 8 LSTM dan Parameter Layer

Pada tahap ini mulai dibangun model *LSTM*. Selain itu, pada tahap ini juga mulai menjalankan parameter *layer*. Seperti yang telah dijelaskan sebelumnya, kombinasi parameter *layer* pada penelitian ini terdiri dari *layer* 2, 3, 4, 5, dan 6 *layer*.

```
optimizer = Adam(learning_rate=0.001)
model.compile(optimizer=optimizer, loss='MSE', metrics=['MAE'])
```

Gbr. 9 *Optimizer* dan *Learning Rate*

Tahap selanjutnya yaitu melakukan kompilasi model menggunakan *optimizer Adam*, serta memasukkan nilai *learning rate* yang digunakan. Adapun nilai *learning rate* yang digunakan dalam penelitian ini adalah *learning rate* 0.001, 0.01, dan 0.1.

```
history = model.fit(X_train, y_train, epochs=50, batch_size=4, validation_data=(X_val, y_val))
```

Gbr. 10 *Epoch* dan *Batch Size*

Code di atas digunakan untuk melatih model dengan parameter *epoch* dan *batch size*. Adapun nilai dari kombinasi parameter *epoch* adalah 10, 20, 30, 40, 50. Nilai kombinasi parameter *batch size* yaitu 4, 8, 16, 32, dan 64. Dari seluruh

kombinasi parameter tersebut akan dilakukan prediksi untuk mendapatkan parameter terbaik yang menghasilkan hasil prediksi yang akurat.

```
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

MAPE = mean_absolute_percentage_error(y_test_actual, predictions)
accuracy = 100 - MAPE
```

Gbr. 11 MAPE dan Akurasi

Metode evaluasi model yang digunakan pada penelitian ini adalah *MAPE* (Mean Absolute Percentage Error). Untuk mengetahui kombinasi parameter yang memiliki hasil prediksi terbaik, maka digunakan evaluasi model menggunakan *MAPE*, dimana *MAPE* akan menampilkan error atau kesalahan prediksi dalam bentuk persentase, sehingga memudahkan dalam mengetahui hasil prediksi mana yang memiliki kesalahan paling rendah. Penggunaan evaluasi model *MAPE* juga memudahkan dalam membaca persentase akurasi dari suatu model. Adapun cara untuk mendapatkan nilai akurasi adalah dengan cara melakukan pengurangan antara 100% dengan nilai persentase dari *MAPE*.

C. Hasil

Parameter yang digunakan dalam membuat prediksi dengan *LSTM* akan mempengaruhi keakuratan hasil prediksi. Oleh karena itu pada penelitian ini, prediksi harga dengan *LSTM* menggunakan kombinasi model parameter yang bervariasi agar dapat menghasilkan hasil prediksi yang akurat. Terdapat lima kombinasi parameter yang digunakan yaitu normalisasi data, pembagian data, *layer*, *batch size*, dan *epoch*. Untuk mendapatkan hasil yang akurat digunakan tiga nilai *learning rate*. Adapun *learning rate* yang digunakan yaitu 0.001, 0.01, dan 0.1.

1) Kombinasi Parameter Metode Normalisasi Data

TABEL VII KOMBINASI PARAMETER METODE NORMALISASI DATA

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	<i>MinMaxScaler</i>	80% : 10% : 10%	2	4	40
2.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	40

Berdasarkan kombinasi parameter metode normalisasi data di atas, didapatkan hasil evaluasi model sebagai berikut.

TABEL VIII

HASIL EVALUASI MODEL PARAMETER METODE NORMALISASI

No.	Metode Normalisasi	Learning Rate	MAPE	Akurasi
1.	<i>MinMaxScaler</i>	0.001	6.10%	93.90%
		0.01	2.47%	97.53%
		0.1	4.58%	95.42%
2.	<i>StandardScaler</i>	0.001	2.53%	97.47%
		0.01	1.43%	98.57%
		0.1	3.10%	96.90%

Hasil evaluasi pada kombinasi parameter metode normalisasi data menunjukkan bahwa metode normalisasi data menggunakan *StandardScaler* memiliki hasil yang lebih akurat dibandingkan dengan metode normalisasi *MinMaxScaler*. *StandardScaler* memiliki hasil akurasi terbaik 98.57% dengan menggunakan *learning rate* 0.001. Sedangkan pada metode normalisasi data *MinMaxScaler* memiliki akurasi terbaik 97.53%.

2) Kombinasi Parameter Pembagian Data

TABEL IX

KOMBINASI PARAMETER PEMBAGIAN DATA

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	40
2.	<i>StandardScaler</i>	70% : 15% : 15%	2	4	40

Berdasarkan kombinasi parameter metode pembagian data di atas, didapatkan hasil evaluasi model sebagai berikut.

TABEL X

HASIL EVALUASI MODEL PARAMETER PEMBAGIAN DATA

No.	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Learning Rate	MAPE	Akurasi
1.	80% : 10% : 10%	0.001	2.53%	97.47%
		0.01	1.43%	98.57%
		0.1	3.10%	96.90%
2.	70% : 15% : 15%	0.001	3.23%	96.77%
		0.01	2.17%	97.83%
		0.1	4.07%	95.93%

Hasil evaluasi pada kombinasi parameter pembagian data di atas, menunjukkan bahwa pembagian data menggunakan 80% data latih, 10% data validasi, dan 10% data uji memiliki hasil yang lebih baik dibandingkan dengan pembagian data 70% data latih, 15% data validasi, dan 15% data uji.

3) Kombinasi Parameter Layer

TABEL XI

KOMBINASI PARAMETER LAYER

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	<i>StandardScaler</i>	80% : 10% : 10%	2	4	40
2.	<i>StandardScaler</i>	80% : 10% : 10%	3	4	40
3.	<i>StandardScaler</i>	80% : 10% : 10%	4	4	40
4.	<i>StandardScaler</i>	80% : 10% : 10%	5	4	40
5.	<i>StandardScaler</i>	80% : 10% : 10%	6	4	40

10%
Berdasarkan kombinasi parameter *layer* di atas, didapatkan hasil evaluasi model sebagai berikut.

TABEL XII
HASIL EVALUASI MODEL PARAMETER *LAYER*

No.	Layer	Learning Rate	MAPE	Akurasi
1.	2	0.001	2.53%	97.47%
		0.01	1.43%	98.57%
		0.1	3.10%	96.90%
2.	3	0.001	2.04%	97.96%
		0.01	2.29%	97.71%
		0.1	4.50%	95.50%
3.	4	0.001	2.09%	97.91%
		0.01	1.89%	98.11%
		0.1	3.61%	96.39%
4.	5	0.001	3.09%	96.91%
		0.01	1.74%	98.26%
		0.1	7.54%	92.46%
5.	6	0.001	2.71%	97.29%
		0.01	2.34%	97.66%
		0.1	11.76%	88.24%

Hasil evaluasi model pada kombinasi parameter *layer* menunjukkan bahwa *layer* 2 memiliki hasil akurasi yang paling akurat dibandingkan dengan jumlah *layer* lainnya.

4) Kombinasi Parameter Batch Size

TABEL XIII
KOMBINASI PARAMETER *BATCH SIZE*

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	StandardScaler	80% : 10% : 10%	2	4	40
2.	StandardScaler	80% : 10% : 10%	2	8	40
3.	StandardScaler	80% : 10% : 10%	2	16	40
4.	StandardScaler	80% : 10% : 10%	2	32	40
5.	StandardScaler	80% : 10% : 10%	2	64	40

Berdasarkan kombinasi parameter *batch size* di atas, didapatkan hasil evaluasi model sebagai berikut.

TABEL XIV
HASIL EVALUASI MODEL PARAMETER *BATCH SIZE*

No.	Batch Size	Learning Rate	MAPE	Akurasi
1.	4	0.001	2.53%	97.47%
		0.01	1.43%	98.57%
		0.1	3.10%	96.90%

2.	8	0.001	2.89%	97.11%
		0.01	2.10%	97.90%
		0.1	2.04%	97.96%
3.	16	0.001	3.73%	96.27%
		0.01	2.36%	97.64%
		0.1	2.03%	97.97%
4.	32	0.001	4.23%	95.77%
		0.01	1.92%	98.08%
		0.1	2.50%	97.50%
5.	64	0.001	4.51%	95.49%
		0.01	2.33%	97.67%
		0.1	2.55%	97.45%

Berdasarkan hasil evaluasi model di atas, dapat diketahui bahwa *batch size* 4 memiliki hasil yang lebih akurat dibandingkan dengan nilai *batch size* lainnya.

5) Kombinasi Parameter Epoch

TABEL XV
KOMBINASI PARAMETER *EPOCH*

No.	Metode Normalisasi	Perbandingan Data (Data Latih : Data Validasi : Data Uji)	Layer	Batch size	Epoch
1.	StandardScaler	80% : 10% : 10%	2	4	10
2.	StandardScaler	80% : 10% : 10%	2	4	20
3.	StandardScaler	80% : 10% : 10%	2	4	30
4.	StandardScaler	80% : 10% : 10%	2	4	40
5.	StandardScaler	80% : 10% : 10%	2	4	50

Berdasarkan kombinasi parameter *epoch* di atas, didapatkan hasil evaluasi model sebagai berikut.

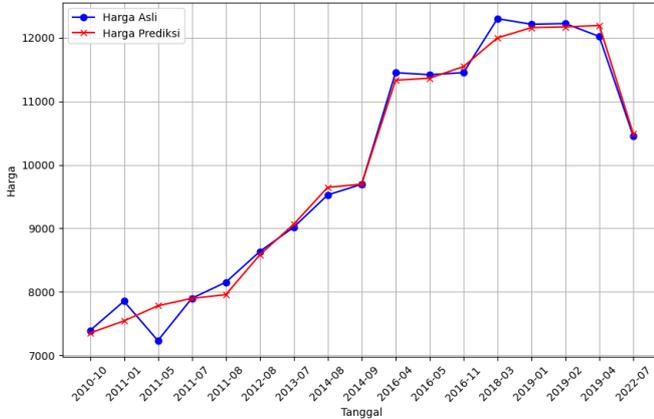
TABEL XVI
HASIL EVALUASI MODEL PARAMETER *EPOCH*

No.	Epoch	Learning Rate	MAPE	Akurasi
1.	10	0.001	4.45%	95.55%
		0.01	2.92%	97.08%
		0.1	3.40%	96.60%
2.	20	0.001	3.53%	96.47%
		0.01	1.77%	98.23%
		0.1	2.31%	97.69%
3.	30	0.001	2.49%	97.51%
		0.01	2.60%	97.40%
		0.1	2.22%	97.78%
4.	40	0.001	2.53%	97.47%
		0.01	1.43%	98.57%
		0.1	3.10%	96.90%

5.	50	0.001	1.90%	98.10%
		0.01	1.94%	98.06%
		0.1	2.92%	97.08%

Hasil evaluasi model pada parameter *epoch* menunjukkan bahwa parameter nilai *epoch* yang memiliki akurasi paling baik ialah *epoch* 40 dengan menggunakan *learning rate* 0.001.

Perbandingan Harga Asli dan Harga Prediksi
Kombinasi 2, Fold 2



Gbr. 12 Grafik Perbandingan Harga Asli dan Harga Prediksi

Grafik di atas adalah grafik perbandingan harga berdasarkan hasil prediksi menggunakan parameter metode normalisasi data *StandardScaler*, pembagian data latih 80%, data validasi 10%, data uji 10%, *layer 2*, *batch size 4*, *epoch 40*, *learning rate 0.01*, dan *optimizer adam*. Gambar grafik perbandingan harga beras asli dengan harga hasil prediksi di atas memperlihatkan bahwa banyak hasil prediksi harga yang sesuai atau mendekati dengan harga aslinya. Hal ini menandakan bahwa hasil prediksi menggunakan parameter tersebut memiliki tingkat kesalahan yang rendah.

Prediksi harga beras dengan berbagai kombinasi parameter yang telah dilakukan mendapatkan nilai akurasi antara 88.24%-98.57% dengan pembagian data secara acak. Dari prediksi yang dilakukan dengan melakukan evaluasi menggunakan *MAPE* untuk mengetahui presentase kesalahan dan akurasi, serta melakukan perbandingan grafik, didapatkan hasil bahwa nilai *MAPE* terbaik 1.43% dan hasil akurasi terbaik ialah 98.57% dengan menggunakan pembagian data secara acak. Dari keseluruhan kombinasi parameter, didapatkan hasil bahwa parameter normalisasi data *StandardScaler*, pembagian data latih 80%, data validasi 10%, data uji 10%, *layer 2*, *batch size 4*, dan *epoch 40* dapat menghasilkan hasil prediksi yang akurat dibandingkan parameter lainnya. Nilai *learning rate* terbaik untuk prediksi berdasarkan penelitian yang telah dilakukan ialah 0.01.

Hasil prediksi menggunakan *LSTM* dengan hasil akurasi terbaik yaitu 98,57% ini akan dibandingkan dengan hasil akurasi prediksi menggunakan *ARIMA* dan *RNN* untuk mengetahui performa dari *LSTM* dalam memprediksi harga beras. Berikut adalah hasil prediksi menggunakan metode *ARIMA*.

TABEL XVII

HASIL PREDIKSI MENGGUNAKAN *ARIMA*

Metode Normalisasi dan Perbandingan Data (Data Latih : Data Validasi : Data Uji)	<i>MAPE</i>	Akurasi
<i>StandardScaler</i> , 80%:10%:10%	10.44%	89.56%
<i>StandardScaler</i> , 70%:15%:15%	10.41%	89.59%
<i>MinMaxScaler</i> , 80%:10%:10%	10.44%	89.56%
<i>MinMaxScaler</i> , 70%:15%:15%	10.41%	89.59%

Berdasarkan tabel tersebut diketahui bahwa nilai akurasi terbaik menggunakan metode *ARIMA* ialah 89.59%. Dari hasil akurasi tersebut dapat disimpulkan bahwa prediksi harga beras menggunakan *LSTM* memiliki hasil yang lebih akurat dengan nilai akurasi 98,57%.

Berikut adalah hasil prediksi menggunakan metode *RNN*.

TABEL XVIII

HASIL PREDIKSI MENGGUNAKAN *RNN*

Metode Normalisasi dan Perbandingan Data (Data Latih : Data Validasi : Data Uji)	<i>MAPE</i>	Akurasi
<i>StandardScaler</i> , 80%:10%:10%	3.53%	96.47%
<i>StandardScaler</i> , 70%:15%:15%	4.05%	95.95%
<i>MinMaxScaler</i> , 80%:10%:10%	5.13%	94.87%
<i>MinMaxScaler</i> , 70%:15%:15%	5.17%	94.83%

Berdasarkan tabel tersebut diketahui bahwa nilai akurasi terbaik menggunakan metode *RNN* ialah 96.47%. Dari hasil akurasi tersebut dapat disimpulkan bahwa prediksi harga beras menggunakan *LSTM* memiliki hasil yang lebih akurat dengan nilai akurasi 98,57%.

Hasil pelatihan model dengan berbagai kombinasi parameter telah dilakukan. Dari pelatihan model tersebut, didapatkan model terbaik dengan parameter metode normalisasi data *StandardScaler*, pembagian data latih 80%, data validasi 10%, data uji 10%, *layer 2*, *batch size 4*, *epoch 40*, dan *learning rate 0.01* dengan hasil akurasi 98.57% menggunakan pembagian data secara acak. Agar hasil prediksi dapat dimanfaatkan oleh banyak kalangan, penulis membuat implementasinya dalam bentuk aplikasi sederhana untuk memprediksi harga beras di Indonesia. Aplikasi sederhana ini dibuat dengan menggunakan bahasa pemrograman *Python* dan *framework Streamlit*. Berikut adalah hasil aplikasi sederhana untuk memprediksi harga beras.



Gbr. 13 Tampilan Aplikasi Prediksi Harga Beras

Gambar di atas, ialah tampilan aplikasi prediksi harga beras. Dalam aplikasi tersebut pengguna dapat menginputkan tanggal untuk memilih di tanggal berapa mereka ingin mengetahui harga prediksinya.



Gbr. 14 Tampilan Input Tanggal

Gambar di atas ialah tampilan Ketika pengguna memilih tanggal prediksi. Pengguna dapat memilih tanggal, bulan, dan tahun yang akan diprediksi.



Gbr. 15 Tampilan Klik Button Prediksi

Gambar di atas ialah tampilan Ketika pengguna telah memilih tanggal dan menjalankan prediksi harga dengan menekan *button* “Prediksi Harga”.



Gbr. 16 Tampilan Prediksi Harga

Gambar di atas merupakan tampilan Ketika aplikasi selesai memproses dan menampilkan prediksi harga sesuai dengan tanggal yang telah dimasukkan oleh pengguna.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa dalam penerapannya, prediksi harga menggunakan *LSTM* akan mendapatkan hasil yang akurat jika menggunakan kombinasi parameter yang bervariasi. Dalam penelitian ini parameter terbaik ialah normalisasi data *StandardScaler*, pembagian data latih 80%, data validasi 10%, data uji 10%, *layer 2*, *batch size 4*, *epoch 40*, dan *learning rate 0.01*. Evaluasi model untuk mengetahui hasil prediksi yang terbaik dalam penelitian ini menggunakan *MAPE*. Evaluasi model dengan *MAPE* dilakukan dengan cara mencari nilai *MAPE* terkecil. Dari hasil perhitungan *MAPE*, dapat diketahui presentase akurasi dari hasil prediksi. Dari penelitian yang dilakukan menunjukkan bahwa performa prediksi harga menggunakan *LSTM* terbukti memiliki hasil yang lebih baik dibandingkan dengan *ARIMA* dan *RNN*, dengan nilai akurasi *LSTM* 98.57%, *ARIMA* 89.59%, dan *RNN* 96.47%.

V. SARAN

Penulis menyadari dalam penelitian ini masih terdapat banyak kekurangan. Salah satu kekurangan dari penelitian ini ialah kurang luasnya dalam mencoba kombinasi parameter untuk mendapatkan hasil yang lebih akurat. Dengan demikian, diharapkan agar penelitian yang dilakukan di masa yang akan datang dapat dilakukan dengan menggunakan parameter yang lebih bervariasi untuk mendapatkan hasil prediksi harga beras yang lebih akurat.

REFERENSI

- [1] V. Arinal and M. Azhari, “Penerapan Regresi Linear Untuk Prediksi Harga Beras Di Indonesia,” *J. Sains dan Teknol.*, vol. 5, no. 1, pp. 341–346, 2023, doi: 10.55338/saintek.v5i1.1417.
- [2] I. Hermawan and E. Budiyantri, “Integrasi Harga Beras Era Perdagangan Terbuka dan Dampaknya Terhadap Swasembada dan Kesejahteraan Pelaku Ekonomi Beras Rice Price Integration in Open Trade Regime and Its Impact on Food Self-Sufficiency and Welfare,” *Bul. Ilm. Litbang Perdagang.*, vol. 14, no. 1, pp. 21–46, 2020, [Online]. Available: <http://www.fao.org/giews/food->
- [3] E. S. Putri and M. Sadikin, “Prediksi Penjualan Produk Untuk Mengestimasi Kebutuhan Bahan Baku Menggunakan Perbandingan Algoritma *LSTM* dan *ARIMA*,” *Format J. Ilm. Tek. Inf.*, vol. 10, no. 2, pp. 162–171, 2021.

- [4] A. Arfan and L. ETP, "Perbandingan Algoritma Long Short-Term Memory dengan SVR Pada Prediksi Harga Saham di Indonesia," *Petir*, vol. 13, no. 1, pp. 33–43, 2020, doi: 10.33322/petir.v13i1.858.
- [5] M. Hussein and Y. Azhar, "Prediksi Harga Minyak Dunia Dengan Metode Deep Learning," *Fountain Informatics J.*, vol. 6, no. 1, pp. 29–34, 2021, doi: 10.21111/fij.v6i1.4446.
- [6] F. I. Sanjaya and D. Heksaputra, "Prediksi Rerata Harga Beras Tingkat Grosir Indonesia dengan Long Short Term Memory," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, no. 2, pp. 163–174, 2020, doi: 10.35957/jatisi.v7i2.388.
- [7] E. Dewi, B. Tarigan, M. F. Balqis, and T. Andri, "Peramalan Harga Beras di Indonesia Dengan ARIMA," vol. 05, no. 02, pp. 117–126, 2024.
- [8] Achmad Sudiro, M. E. . C. P. H. . Prof. Dr. Drs. Achmad Sudiro, and Oktaria Ardika Putri, *Manajemen Sumber Daya Manusia*. Bumi Aksara, 2022.
- [9] P. Wahono, D. Mugia, B. Rachman, and S. R. Widiyanto, "Integrasi Data Kontak HP Berbasis Kartu SIM Menggunakan Aplikasi atau Platform Lain," *Semin. Nas. Teknol. Komput. Sains*, vol. 1, no. 1, pp. 44–50, 2020, [Online]. Available: <http://seminar-id.com/prosiding/index.php/sainteks/article/view/403>
- [10] N. A. Permatasari, Y. H. Chrisnanto, and A. K. Ningsih, "Segmentasi Kasus Data Kematian Covid 19 Di Jawa Barat Menggunakan Algoritma DBSCAN," *IJESPG (International J. Eng. Econ. Soc. Polit. Gov.)*, vol. 1, no. 4, pp. 119–128, 2023.
- [11] A. O. Praneswara, "Perbandingan K-Nearest Neighbors, Support Vector Dan Random Forest Pada Prediksi Medical Cost," *Indones. J. Comput. Sci.*, vol. 12, no. 4, pp. 2035–2048, 2023.
- [12] V. Ariyani, P. Putri, A. B. Prasetyo, and D. Eridani, "Perbandingan Kinerja Algoritme Naive Bayes Dan K-Nearest Neighbor (KNN) Untuk Prediksi Harga Rumah," *J. Ilm. Tek. Elektro*, vol. 24, no. 4, pp. 162–171, 2022, [Online]. Available: <https://ejournal.undip.ac.id/index.php/transmisi>
- [13] F. N. Iman and D. Wulandari, "Prediksi Harga Saham Menggunakan Metode Long Short Term Memory (LSTM)," *J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 3, pp. 601–616, 2023, [Online]. Available: <https://journal.mediapublikasi.id/index.php/logic/article/view/1855/1497>
- [14] V. Gallan, "LSTM (Long Short Term Memory)," Bina Nusantara IT Division. Accessed: Mar. 16, 2024. [Online]. Available: <https://medium.com/bina-nusantara-it-division/lstm-long-short-term-memory-d29779e2ebf8>
- [15] A. D. Milniadi and N. O. Adiwijaya, "Analisis Perbandingan Model Arima Dan Lstm Dalam Peramalan Harga Penutupan Saham (Studi Kasus : 6 Kriteria Kategori Saham Menurut Peter Lynch)," *SIBATIK J. J. Ilm. Bid. Sos. Ekon. Budaya, Teknol. dan Pendidik.*, vol. 2, no. 6, pp. 1683–1692, 2023, doi: 10.54443/sibatik.v2i6.798.
- [16] BPS (Badan Pusat Statistik), "Rata-rata Harga Beras di Tingkat Perdagangan Besar (Grosir) Indonesia (Perusahaan)." Accessed: Mar. 25, 2024. [Online]. Available: <https://www.bps.go.id/id/statistics-table/2/Mjk1IzI=/rata-rata-harga-beras-di-tingkat-perdagangan-besar-grosir-indonesia.html>