

Implementasi Deep Neural Network untuk Prediksi Harga Saham PT Bank Central Asia Tbk

Muhammad Rakha Almasah*, Wahyu Aji Eko Prabowo

Fakultas Ilmu Komputer, Program Studi PJJ Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia

Email: ¹*118202200017@mhs.dinus.ac.id, ²prabowo@dsn.dinus.ac.id

Email Penulis Korespondensi: 118202200017@mhs.dinus.ac.id

Submitted 21-04-2025; Accepted 30-04-2025; Published 30-04-2025

Abstrak

Prediksi harga saham merupakan salah satu tantangan utama dalam analisis pasar keuangan. Penelitian ini mengembangkan model Deep Neural Network (DNN) menggunakan TensorFlow untuk memprediksi harga saham PT Bank Central Asia Tbk (BBCA) berdasarkan data historis harga saham, Indeks Harga Saham Gabungan (IHSG), dan nilai tukar USD/IDR. Model dioptimalkan melalui hyperparameter tuning menggunakan KerasTuner dengan metode Hyperband, yang memungkinkan eksplorasi kombinasi hyperparameter secara lebih efisien. Dari hasil tuning, diperoleh konfigurasi model terbaik dengan learning rate 0.001269, empat hidden layer, serta penerapan Batch Normalization, regularisasi kernel L2, dan dropout pada setiap lapisan tersembunyi. Evaluasi model dilakukan dengan berbagai batch size (4, 16, 32, 64, dan 128) menggunakan metrik Mean Squared Error (MSE) dan Mean Absolute Error (MAE). Hasil menunjukkan bahwa batch size 128 memberikan performa terbaik, dengan nilai MSE 0.0094 dan MAE 0.0490 pada data uji, yang mengindikasikan akurasi tinggi dan kemampuan generalisasi model terhadap data yang tidak dilibatkan dalam pelatihan. Model terbaik kemudian diimplementasikan sebagai API menggunakan Flask, di-deploy ke Google Cloud Run, dan diintegrasikan dengan aplikasi mobile berbasis Flutter. Pengujian menunjukkan bahwa API dapat menangani permintaan (request) dengan cepat menggunakan TensorFlow dan menghasilkan prediksi yang akurat. Dengan demikian, penelitian ini berhasil mengembangkan sistem prediksi harga saham BBCA berbasis DNN yang dapat diterapkan dalam pengambilan keputusan investasi berbasis data.

Kata Kunci: Deep Neural Network; TensorFlow; Hyperparameter Tuning; Prediksi Harga Saham; KerasTuner; Cloud Run; Flutter

Abstract

Stock price prediction is one of the main challenges in financial market analysis. This study develops a Deep Neural Network (DNN) model using TensorFlow to predict the stock price of PT Bank Central Asia Tbk (BBCA) based on historical stock price data, the Jakarta Composite Index (IHSG), and the USD/IDR exchange rate. The model is optimized through hyperparameter tuning using KerasTuner with the Hyperband method, allowing for more efficient exploration of hyperparameter combinations. The tuning process yielded the best model configuration with a learning rate of 0.001269, four hidden layers, and the application of Batch Normalization, L2 kernel regularization, and dropout in each hidden layer. The model was evaluated using various batch sizes (4, 16, 32, 64, and 128) with Mean Squared Error (MSE) and Mean Absolute Error (MAE) as performance metrics. The results show that batch size 128 provides the best performance, with MSE of 0.0094 and MAE of 0.0490 on the test data, indicating high accuracy and good generalization on unseen data. The best model was then implemented as an API using Flask, deployed to Google Cloud Run, and integrated with a Flutter-based mobile application. Testing confirmed that the API can handle requests quickly using TensorFlow and produce accurate predictions. Thus, this study successfully developed a DNN-based BBCA stock price prediction system that can be applied to data-driven investment decision-making.

Keywords: Deep Neural Network; TensorFlow; Hyperparameter Tuning; Stock Price Prediction; KerasTuner; Cloud Run; Flutter

1. PENDAHULUAN

Saham merupakan instrumen keuangan yang diterbitkan oleh perusahaan sebagai bukti kepemilikan investor atas bagian tertentu dari perusahaan [1]. Melalui investasi saham, investor berpeluang memperoleh keuntungan dalam bentuk dividen dan/atau *capital gain*. Namun, selain potensi keuntungan, investasi saham juga memiliki risiko, seperti *capital loss* dan likuidasi perusahaan, yang dapat menyebabkan kerugian bagi investor. Risiko *capital gain* dan *capital loss* terjadi karena harga saham bersifat fluktuatif [2].

Di Indonesia, kinerja keseluruhan saham yang tercatat di bursa dapat diukur melalui Indeks Harga Saham Gabungan (IHSG), yang mencerminkan fluktuasi harga saham secara agregat di pasar modal. Pergerakan IHSG dipengaruhi oleh berbagai faktor makroekonomi, di antaranya tingkat inflasi, suku bunga Bank Indonesia, dan nilai tukar USD/IDR, yang telah terbukti berkontribusi terhadap fluktuasi pasar saham [3]. Faktor-faktor makroekonomi tersebut tidak hanya memengaruhi IHSG tetapi juga berdampak pada pergerakan harga saham individu yang tergabung dalam indeks tersebut. Selain itu, IHSG menggunakan metode pembobotan berdasarkan kapitalisasi pasar, di mana perusahaan dengan kapitalisasi pasar terbesar memiliki bobot yang lebih tinggi dalam indeks. Akibatnya, saham-saham berkapitalisasi besar lebih sensitif terhadap perubahan faktor makroekonomi dan memiliki pengaruh yang lebih besar terhadap pergerakan IHSG secara keseluruhan [4].

Salah satu perusahaan dengan kapitalisasi pasar terbesar di Indonesia adalah PT Bank Central Asia Tbk (BBCA) [5]. Sebagai perusahaan dengan kapitalisasi pasar terbesar, pergerakan harga saham BBCA menjadi perhatian utama bagi investor. Hal ini disebabkan oleh kondisi fundamental BBCA yang secara konsisten menunjukkan pertumbuhan pendapatan setiap tahun, sehingga meningkatkan kepercayaan investor dalam berinvestasi pada saham perusahaan tersebut serta mendorong kenaikan harga saham BBCA [6]. Kondisi ini pada akhirnya memperkuat posisi BBCA sebagai perusahaan dengan kapitalisasi pasar terbesar di Indonesia. Mengingat pentingnya pergerakan harga saham BBCA bagi

investor, diperlukan sistem prediksi harga saham yang mampu menganalisis pola pergerakan harga secara lebih akurat dan berbasis data. Sistem ini diharapkan dapat membantu investor dalam mengambil keputusan investasi yang lebih terinformasi dan strategis.

Berbagai penelitian terdahulu telah dilakukan untuk memprediksi harga saham BBKA. Salah satu penelitian menggunakan metode *Long Short-Term Memory* (LSTM) dan membandingkannya dengan *Support Vector Regression* (SVR) dengan data harga penutupan saham BBKA dari 22 Oktober 2018 hingga 20 Oktober 2023. Hasil penelitian menunjukkan bahwa model LSTM memberikan performa prediksi yang lebih baik dibandingkan SVR, dengan *Mean Absolute Percentage Error* (MAPE) sebesar 1.07% [7]. Penelitian lain menerapkan algoritma *Deep Learning Gated Recurrent Unit* (GRU) untuk memprediksi harga saham BBKA, dengan data yang mencakup harga pembukaan, penutupan, tertinggi, terendah, serta volume perdagangan dalam rentang waktu 2 Januari 2019 hingga 19 Maret 2024. Model ini dilatih menggunakan skenario pembagian data sebesar 80:20 dan menghasilkan *Root Mean Squared Error* (RMSE) sebesar 1.039, *Mean Squared Error* (MSE) sebesar 1.079, *Mean Absolute Error* (MAE) sebesar 0.842, *R-Squared* sebesar 0.983, *Mean Geometric Deviation* (MGD) sebesar 0.0037, dan *Mean Percentage Deviation* (MPD) sebesar 0.0197. Selain itu, model ini juga menghasilkan akurasi sebesar 54.87%, recall sebesar 59.23%, F1-score sebesar 58.11%, dan precision sebesar 57.03% [8]. Penelitian lainnya juga menggunakan LSTM untuk memprediksi harga saham BBKA dengan data harga terendah dalam periode 2 Januari 2020 hingga 30 Oktober 2023. Model LSTM yang dioptimalkan menggunakan *Adam* menghasilkan RMSE sebesar 40.85, MAPE sebesar 0.71%, dan MSE sebesar 6662.76 [9]. Meskipun berbagai metode telah diterapkan dalam penelitian terdahulu, sebagian besar studi hanya mempertimbangkan data historis harga saham tanpa memasukkan faktor makroekonomi seperti IHSG dan nilai tukar USD/IDR, sehingga model yang dikembangkan kurang aplikatif jika diterapkan langsung oleh investor dalam pengambilan keputusan investasi yang lebih komprehensif.

Penelitian lain membahas prediksi harga saham BBKA setelah *stock split* dengan menggunakan metode *Artificial Neural Network* (ANN) berbasis algoritma *backpropagation*. Penelitian ini menggunakan data variabel lag saham BBKA dan nilai tukar USD/IDR dalam periode mingguan dari 17 Oktober 2021 hingga 17 September 2023. Hasil penelitian menunjukkan bahwa model ANN menghasilkan *Root Mean Squared Error* (RMSE) sebesar 121.16 dan *Mean Absolute Percentage Error* (MAPE) sebesar 1.12%. Namun, penelitian ini hanya menggunakan satu *hidden layer* dan belum mempertimbangkan faktor makroekonomi seperti IHSG, serta belum menerapkan teknik *hyperparameter tuning* yang lebih kompleks, seperti optimalisasi *learning rate*, *batch size*, dan parameter lainnya [5].

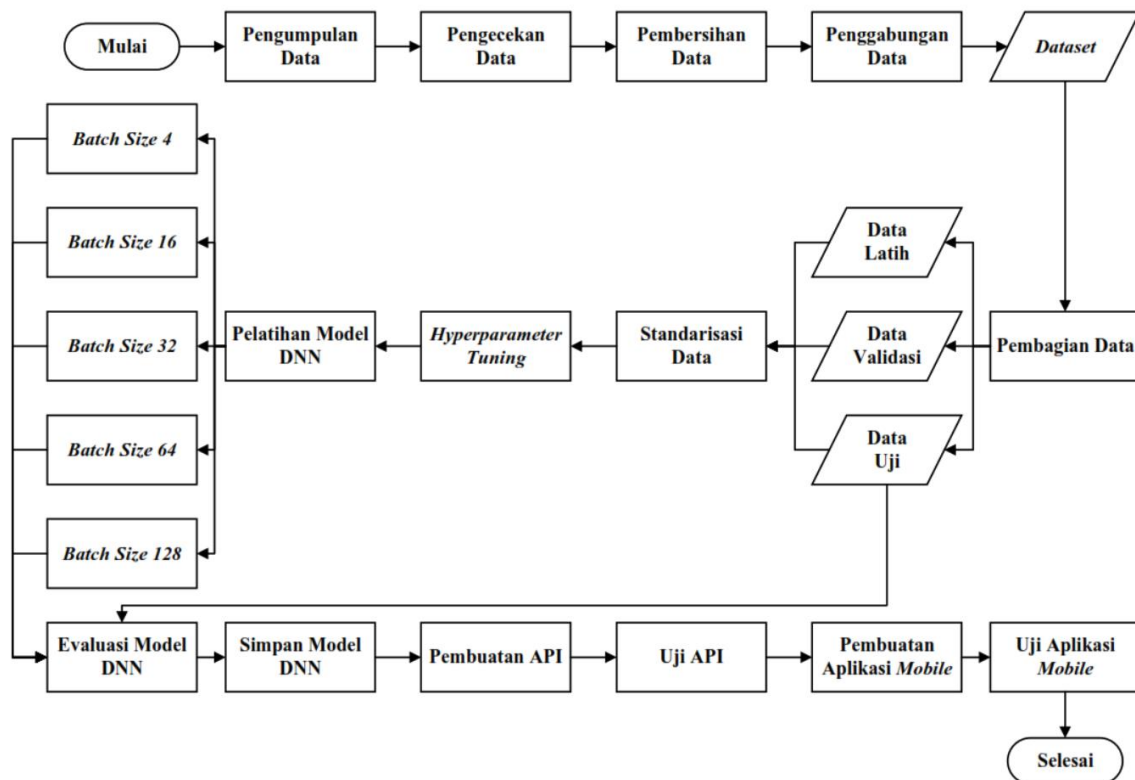
Di sisi lain, penelitian lain mengkaji prediksi suhu air sungai per jam berdasarkan data meteorologi. Studi ini menggunakan variabel independen berupa suhu udara (°C), kelembaban relatif (%), tekanan stasiun (mm), dan kecepatan angin (m/s), dengan variabel dependen suhu air (°C). Penelitian ini menguji beberapa algoritma, termasuk regresi linear dengan satu lapisan dan multilapisan, serta *Deep Neural Network* (DNN). Hasil penelitian menunjukkan bahwa algoritma DNN multilapisan memiliki performa yang lebih unggul dibandingkan model lainnya, dengan peningkatan koefisien determinasi sebesar 26% dibandingkan model LR satu lapisan (model dasar) dan 12% lebih tinggi dibandingkan model LR multilapisan [10].

Berdasarkan penelitian terdahulu yang menunjukkan keunggulan model DNN dalam prediksi suhu air sungai, penelitian ini mengembangkan model DNN berbasis *TensorFlow* untuk memprediksi harga saham BBKA. Studi literatur yang telah dilakukan menunjukkan bahwa sebagian besar penelitian sebelumnya hanya berfokus pada data historis harga saham tanpa memperhitungkan faktor makroekonomi seperti IHSG dan nilai tukar USD/IDR sebagai variabel prediksi. Selain itu, beberapa penelitian menggunakan metode *deep learning* tanpa penerapan teknik *hyperparameter tuning* yang optimal. Model yang dikembangkan juga cenderung sederhana, seperti penggunaan satu *hidden layer* atau parameter yang tidak dioptimasi. Berdasarkan temuan tersebut, terdapat gap penelitian dalam hal pemanfaatan faktor makroekonomi sebagai variabel tambahan, penerapan teknik *hyperparameter tuning* untuk optimasi model, serta pengembangan arsitektur DNN yang lebih kompleks guna meningkatkan akurasi prediksi harga saham. Kebaruan penelitian ini terletak pada penggunaan data harian dalam rentang waktu 8 Juni 2004 hingga 5 Maret 2025, serta pengembangan model yang lebih kompleks dengan menerapkan *batch normalization* dan *dropout layer*. Selain itu, dilakukan *hyperparameter tuning* pada *learning rate*, *kernel regularizer* (L2), *dropout rate*, serta jumlah *hidden units* dengan berbagai percobaan *batch size*. Variabel yang digunakan dalam penelitian ini mencakup harga pembukaan saham BBKA, harga pembukaan IHSG, dan harga pembukaan nilai tukar USD/IDR untuk memprediksi harga penutupan saham BBKA. Dengan pendekatan ini, model DNN yang dikembangkan diharapkan lebih efektif dan aplikatif bagi investor dalam melakukan prediksi harga saham BBKA secara harian, sehingga memungkinkan investor untuk menentukan apakah harga saham BBKA akan mengalami kenaikan atau penurunan berdasarkan hasil prediksi. Selain itu, penelitian ini juga akan dikembangkan dalam bentuk aplikasi *mobile* berbasis *Flutter*, di mana model DNN dalam *SavedModel* format akan disimpan dan di-deploy menggunakan *Flask* dan *Docker* ke *Google Cloud Run*, kemudian diintegrasikan dengan *Artificial Programming Interface* (API) prediksi untuk meningkatkan aksesibilitas pengguna.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian yang dilakukan dalam penelitian ini dijelaskan secara umum pada Gambar 1.



Gambar 1. Tahapan Penelitian

Gambar 1 menggambarkan alur tahapan penelitian yang dimulai dari proses pengumpulan, pengecekan, pembersihan, hingga penggabungan data menjadi satu *dataset* yang utuh. *Dataset* tersebut kemudian dibagi menjadi tiga bagian, yaitu data latih, data validasi, dan data uji. Tahap selanjutnya adalah standarisasi data menggunakan *Standard Scaler* untuk memastikan keseragaman skala antar fitur. Kemudian dilakukan proses *hyperparameter tuning* untuk memperoleh konfigurasi model DNN terbaik. Model tersebut dilatih ulang menggunakan lima variasi *batch size* (4, 16, 32, 64, dan 128), yang masing-masing dievaluasi untuk menentukan *batch size* dengan performa terbaik berdasarkan metrik MSE dan MAE pada data uji. Setelah model terbaik dipilih, model disimpan dalam format *SavedModel* dan diintegrasikan ke dalam API berbasis *Flask*. API tersebut di-deploy ke *Google Cloud Run* dan diuji untuk memastikan fungsionalitasnya. Tahap akhir adalah pembuatan dan pengujian aplikasi *mobile* berbasis *Flutter* yang terhubung ke API, sehingga pengguna dapat melakukan prediksi harga saham BBKA secara *real-time* melalui antarmuka aplikasi.

2.2 Pengumpulan Data

Yahoo Finance API merupakan pustaka *Python* yang banyak digunakan untuk menyediakan data keuangan secara *real-time* maupun historis, mencakup berbagai aset seperti saham, dana investasi, obligasi, dan komoditas [11]. Pada tahap ini, data yang digunakan mencakup harga saham BBKA, IHSG, dan nilai tukar USD/IDR yang diperoleh melalui *Yahoo Finance* API. Data tersebut dikumpulkan dalam rentang waktu 8 Juni 2004 hingga 5 Maret 2025 untuk memastikan cakupan historis yang luas dalam analisis prediksi harga saham.

2.3 Pengecekan Data

Pandas adalah pustaka *data science* berbasis *NumPy* yang mendukung operasi matematis efisien dan menyediakan *wrapper Matplotlib* untuk visualisasi data [12]. Pada tahap ini, dilakukan pengecekan terhadap data yang telah dikumpulkan menggunakan *Pandas*, yang mencakup pemeriksaan informasi tiap kolom, deteksi nilai yang hilang (*missing values*), identifikasi data duplikat, analisis statistik deskriptif, dan visualisasi data.

2.4 Pembersihan Data

Interpolasi linear merupakan salah satu metode yang digunakan untuk menangani *missing values* dalam proses pembersihan data. Metode ini bekerja dengan memperkirakan nilai data yang hilang berdasarkan dua titik data terdekat dalam urutan data satu dimensi, sehingga menghasilkan estimasi nilai yang lebih akurat [13]. Dalam penelitian ini, interpolasi linear diterapkan pada data harga pembukaan nilai tukar USD/IDR untuk menangani nilai yang kurang dari 1000, yang dianggap sebagai data tidak valid atau anomali. Data dengan nilai kurang dari batas tersebut terlebih dahulu diubah menjadi *NaN* sebelum dilakukan interpolasi secara *forward direction* untuk memperkirakan nilai yang sesuai berdasarkan pola data yang ada.

2.5 Penggabungan Data

Setelah proses pembersihan data, tahap selanjutnya adalah penggabungan data menjadi satu *dataset* yang terdiri dari variabel *Date*, *Open_BBCA*, *Close_BBCA*, *Open_USDIDR*, dan *Open_IHSG*. *Dataset* yang telah digabungkan tersebut dapat dilihat pada Tabel 1, yang menampilkan cuplikan beberapa contoh data historis yang digunakan dalam penelitian ini.

Tabel 1. Sampel *Dataset*

<i>Date</i>	<i>Open_BBCA</i>	<i>Close_BBCA</i>	<i>Open_USDIDR</i>	<i>Open_IHSG</i>
2004-06-08	175.0	177.5	9294.2998046875	705.7849731445312
2004-06-09	177.5	180.0	9241.2001953125	704.0499877929688
2004-06-10	180.0	180.0	9280.400390625	706.4819946289062
2004-06-11	177.5	180.0	9349.400390625	706.3419799804688
2004-06-14	180.0	177.5	9428.599609375	702.489013671875
2004-06-15	175.0	182.5	9374.400390625	696.1320190429688
2004-06-16	182.5	180.0	9387.2998046875	710.6199951171875
2004-06-17	177.5	177.5	9412.2001953125	706.7039794921875
...	...	...	...	...
2025-02-24	8925.0	8925.0	16278.0	6801.67578125
2025-02-25	8900.0	8825.0	16279.2998046875	6749.43701171875
2025-02-26	8800.0	8775.0	16314.0	6627.02099609375
2025-02-27	8800.0	8525.0	16367.400390625	6609.88818359375
2025-02-28	8525.0	8425.0	16553.69921875	6446.64404296875
2025-03-03	8600.0	8800.0	16522.0	6362.14892578125
2025-03-04	8900.0	8850.0	16451.19921875	6524.701171875
2025-03-05	8825.0	9000.0	16379.0	6404.68896484375

Dataset pada Tabel 1 mencakup lima variabel utama, yaitu tanggal perdagangan (*Date*), harga pembukaan saham BBCA (*Open_BBCA*), harga penutupan saham BBCA (*Close_BBCA*), harga pembukaan nilai tukar USD terhadap IDR (*Open_USDIDR*), dan harga pembukaan IHSG (*Open_IHSG*). Dalam penelitian ini, variabel *Open_BBCA*, *Open_USDIDR*, dan *Open_IHSG* digunakan sebagai variabel independen (fitur *input*), sedangkan *Close_BBCA* berfungsi sebagai variabel dependen (target *output*) yang akan diprediksi oleh model.

2.6 Pembagian Data

Pada tahap ini, *dataset* dibagi menjadi tiga bagian utama, yaitu data latih, data validasi, dan data uji. Data latih digunakan untuk melatih model *deep learning* agar dapat menjalankan fungsi tertentu. Data validasi berfungsi untuk mengevaluasi performa model selama pelatihan serta membantu dalam penyempurnaan arsitektur model (*fine-tuning*). Sementara itu, data uji digunakan untuk evaluasi akhir guna mengukur kemampuan model dalam memprediksi data yang belum pernah digunakan sebelumnya [14]. Proporsi pembagian *dataset* tersebut dapat dilihat pada Tabel 2, yang merangkum jumlah data untuk setiap kategori.

Tabel 2. Pembagian Data

Jenis Data	Jumlah Data	Persentase (%)
Data Latih	4010	80
Data Validasi	501	10
Data Uji	502	10
Total	5013	100

Sebagaimana ditunjukkan pada Tabel 2, sebanyak 80% data dialokasikan untuk pelatihan model, 10% untuk validasi model selama proses *tuning*, dan 10% sisanya untuk pengujian akhir model. Strategi pembagian data ini bertujuan untuk memastikan bahwa model dapat belajar dari sebagian besar data, melakukan *tuning* parameter berdasarkan data validasi, serta dievaluasi menggunakan data uji yang benar-benar baru, sehingga memberikan ukuran performa model yang objektif.

2.7 Standarisasi Data

Standard Scaler merupakan salah satu metode yang digunakan untuk proses standarisasi *dataset*. *Standard Scaler* adalah teknik standarisasi data, juga dikenal sebagai *Z-score normalization*, yang mentransformasikan distribusi fitur agar memiliki rata-rata 0 dan simpangan baku 1. Proses ini dilakukan dengan mengurangi setiap nilai data dengan rata-rata (*mean*) fitur tersebut, kemudian membaginya dengan simpangan baku fitur yang sama. Dengan demikian, hasil transformasi menghasilkan distribusi dengan nilai tengah 0 dan varians 1, sehingga skala semua fitur menjadi seragam. Standarisasi ini penting untuk memastikan setiap fitur memiliki kontribusi yang seimbang dalam pelatihan model serta menghindari dominasi fitur tertentu akibat perbedaan skala [15]. Dalam penelitian ini, *Standard Scaler* diterapkan pada data latih, validasi, dan uji sebelum proses pemodelan. Namun, data uji hanya digunakan untuk evaluasi akhir dan tidak dilibatkan dalam proses *hyperparameter tuning* dan pelatihan model.

2.8 Hyperparameter Tuning

Hyperparameter tuning adalah proses pencarian kombinasi *hyperparameter* optimal untuk meningkatkan performa model statistik. Proses ini dilakukan dengan melatih model menggunakan berbagai nilai *hyperparameter* dalam rentang yang telah ditentukan, kemudian mengevaluasi performanya pada data validasi. *Hyperparameter* dengan hasil terbaik dalam prediksi terhadap data di luar sampel (*out-of-sample*), berdasarkan metrik evaluasi yang dipilih, akan digunakan sebagai konfigurasi akhir model [16]. Dalam penelitian ini, model *Deep Neural Network* (DNN) dirancang menggunakan arsitektur *multi-layer perceptron* (MLP) dengan empat lapisan tersembunyi (*hidden layers*). Untuk mendapatkan konfigurasi terbaik, model ini menggunakan *hyperparameter tuning* dengan *KerasTuner* menggunakan metode *Hyperband*. *Hyperband* adalah salah satu algoritma *tuning* yang digunakan untuk mengeksplorasi kombinasi *hyperparameter* secara efisien. Algoritma ini bekerja dengan mengalokasikan sumber daya secara adaptif, di mana model dengan performa buruk dieliminasi lebih awal, sehingga hanya model dengan konfigurasi terbaik yang dievaluasi secara lebih mendalam. Dengan pendekatan ini, *Hyperband* dapat menghemat waktu komputasi dibandingkan pencarian acak atau *grid search* [17]. Pada setiap lapisan tersembunyi, jumlah *neuron* ditentukan secara dinamis sebagai *hyperparameter*, dengan rentang antara 32 hingga 512 neuron, bertambah dengan langkah 32 unit. Model ini menggunakan fungsi aktivasi ReLU (*Rectified Linear Unit*) untuk menangkap pola *non-linear* secara efektif [18]. Untuk mengatasi *overfitting*, model menerapkan regularisasi L2 (*L2 kernel regularization*) dengan nilai yang disesuaikan dalam rentang $1e-5$ hingga $1e-1$, serta *Dropout layer* dengan nilai antara 0.1 hingga 0.5, yang membantu meningkatkan generalisasi model dengan mengabaikan sebagian unit selama pelatihan [19]. Selain itu, *Batch Normalization* diterapkan setelah setiap lapisan tersembunyi untuk mempercepat pelatihan dan menjaga kestabilan distribusi aktivasi [20]. Lapisan keluaran (*output layer*) menggunakan satu *neuron* dengan fungsi aktivasi linear, yang sesuai dengan tugas prediksi regresi dalam penelitian ini. Model ini dikompilasi menggunakan optimasi *AdamW*, dengan *learning rate* yang ditentukan dalam rentang $1e-5$ hingga $1e-2$. Sebagai fungsi *loss*, digunakan *Mean Squared Error* (MSE) untuk mengoptimalkan model, sedangkan *Mean Absolute Error* (MAE) digunakan sebagai metrik evaluasi.

2.9 Pelatihan Model DNN

Pada tahap ini, hasil terbaik dari proses *hyperparameter tuning* digunakan untuk melatih ulang model dengan lima variasi ukuran *batch*, yaitu 4, 16, 32, 64, dan 128. *Batch size* mengacu pada jumlah sampel yang diberikan kepada *optimizer* sebelum pembaruan bobot dilakukan dalam proses pelatihan, yang dapat memengaruhi efisiensi komputasi dan konvergensi model [19]. Model kemudian dilatih selama 1000 *epochs* menggunakan *TensorFlow*, dengan *AdamW* sebagai *optimizer*. Pemilihan model terbaik didasarkan pada nilai *validation loss* terendah yang diperoleh selama proses pelatihan.

2.10 Evaluasi Model DNN

Pada tahap ini, evaluasi model dilakukan menggunakan dua metrik utama, yaitu *Mean Squared Error* (MSE) dan *Mean Absolute Error* (MAE). MSE mengukur rata-rata kesalahan kuadrat antara nilai prediksi dan nilai aktual, di mana nilai MSE yang lebih kecil menunjukkan prediksi yang lebih akurat. Sementara itu, MAE mengukur rata-rata besarnya kesalahan prediksi tanpa memperhitungkan arah kesalahan [21]. Kedua metrik ini digunakan untuk mengevaluasi hasil pelatihan pada seluruh variasi *batch size* menggunakan data uji. Model dengan nilai MSE dan MAE terendah akan dipilih sebagai model terbaik untuk digunakan dalam proses prediksi.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (1)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y(i) - \hat{y}(i)| \quad (2)$$

2.11 Pembuatan dan Uji API

Pada tahap ini, model terbaik yang telah dipilih disimpan dalam format *SavedModel*. Selanjutnya, dilakukan pembuatan API prediksi menggunakan *Flask*, yang kemudian di-deploy ke *Google Cloud Platform* (GCP) melalui *Google Cloud Run* menggunakan *Docker* sebagai metode *deployment* berbasis *container*. Pendekatan ini terbukti efektif dan efisien untuk menyajikan layanan prediktif berbasis *machine learning* secara skalabel tanpa memerlukan infrastruktur *Graphics Processing Unit* (GPU), serta mendukung otomatisasi penskalaan dan efisiensi biaya dalam lingkungan *serverless* [22]. Selain itu, penggunaan *Flask* dalam pengembangan layanan prediktif yang dikemas dalam *Docker* dan di-deploy melalui *Google Cloud Run* juga telah berhasil digunakan dalam implementasi sistem *machine learning* interaktif yang berbasis antarmuka *web*, menunjukkan bahwa pendekatan ini mendukung pengembangan sistem prediksi yang dapat diakses publik melalui *cloud* secara andal dan fleksibel [23]. Setelah proses *deployment* selesai, dilakukan pengujian API dengan mengirimkan permintaan (*request*) dalam format JSON, yang berisi data masukan berupa *Open BBKA*, *Open USD/IDR*, dan *Open IHSG*. API yang telah dikembangkan harus mampu mengembalikan hasil prediksi berupa nilai *Close BBKA*, yang akan digunakan sebagai keluaran sistem.

2.12 Pembuatan dan Uji Aplikasi Mobile

Setelah API selesai dibuat, tahap berikutnya adalah pengembangan aplikasi *mobile* menggunakan *framework Flutter*, di mana aplikasi ini berfungsi sebagai *Front-End*, sementara *Back-End* untuk prediksi saham menggunakan API yang telah dikembangkan sebelumnya. *Flutter* dipilih karena kemampuannya membangun aplikasi lintas platform (*Android* dan *iOS*)

hanya dengan satu basis kode, serta didukung oleh berbagai paket seperti *http* untuk melakukan komunikasi dengan API melalui metode *REST* [24]. Pembuatan aplikasi mencakup antarmuka pengguna (UI) yang memungkinkan pengguna memasukkan nilai *Open BBCA*, *Open USD/IDR*, dan *Open IHSG*. Selain itu, terdapat dua tombol utama, yaitu *Predict* dan *Clear*. Jika pengguna memilih tombol *Predict*, data yang dimasukkan akan dikirim ke API, yang kemudian mengembalikan hasil prediksi harga saham. Sebaliknya, jika tombol *Clear* dipilih, maka seluruh masukan pengguna akan dihapus. Penggunaan *Flutter* sebagai antarmuka klien dalam model ini terbukti efektif untuk membangun sistem *mobile* yang ringan dan cepat terhubung dengan *RESTful API* secara profesional dan fleksibel [24].

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dan analisis dari proses *hyperparameter tuning*, evaluasi model dengan variasi *batch size*, dan akurasi dari hasil prediksi model terbaik pada data uji. Model dengan akurasi terbaik selanjutnya digunakan dalam tahap *deployment* untuk implementasi sistem prediksi harga saham BBKA.

3.1 Hasil Hyperparameter Tuning

Pada tahap *hyperparameter tuning*, dilakukan pencarian konfigurasi terbaik untuk model *Deep Neural Network* (DNN) guna memperoleh performa optimal dalam prediksi harga saham BBKA. Proses *tuning* ini menggunakan *KerasTuner* dengan algoritma *Hyperband*, yang memungkinkan eksplorasi kombinasi *hyperparameter* secara efisien dengan mengalokasikan sumber daya lebih banyak pada model yang menunjukkan performa lebih baik. Hasil *tuning* menunjukkan bahwa konfigurasi *hyperparameter* terbaik diperoleh dengan *best validation loss* sebesar 0.0188, dengan total waktu pemrosesan selama 2 jam 37 menit 17 detik. Rincian konfigurasi *hyperparameter* terbaik yang diperoleh ditampilkan pada Tabel 3.

Tabel 3. Hasil Hyperparameter Tuning Terbaik

Hyperparameter	Nilai Optimal
Best Validation Loss	0.0188
Total Waktu Pemrosesan	2 jam 37 menit 17 detik
Learning Rate	0.001269
Units (Layer 1)	512
Kernel Regularizer (Layer 1)	0.000719
Dropout (Layer 1)	0.5
Units (Layer 2)	64
Kernel Regularizer (Layer 2)	0.000628
Dropout (Layer 2)	0.2
Units (Layer 3)	512
Kernel Regularizer (Layer 3)	0.002350
Dropout (Layer 3)	0.3
Units (Layer 4)	256
Kernel Regularizer (Layer 4)	0.000178
Dropout (Layer 4)	0.4

Tabel 3 menunjukkan bahwa konfigurasi terbaik diperoleh pada model dengan 4 *hidden layers*, dengan jumlah *neuron* optimal sebesar 512 pada lapisan pertama dan ketiga, sementara lapisan kedua dan keempat menggunakan 64 dan 256 *neuron*. Regularisasi *kernel L2* diterapkan pada setiap lapisan dengan nilai kecil yang bervariasi antara 0.000178 hingga 0.002350 untuk mengontrol kompleksitas model dan mencegah *overfitting*. Selain itu, *dropout* diterapkan dengan kisaran nilai antara 0.2 hingga 0.5, bertujuan untuk meningkatkan kemampuan generalisasi model. *Learning rate* optimal sebesar 0.001269 digunakan dalam optimasi model untuk memastikan proses pembelajaran berjalan stabil. Hasil *hyperparameter tuning* ini akan digunakan dalam tahap pelatihan model dengan variasi *batch size*, guna mengevaluasi dampak ukuran *batch* terhadap performa model dalam memprediksi harga saham BBKA.

3.2 Evaluasi Pengaruh Batch Size terhadap Performa Model

Setelah memperoleh konfigurasi *hyperparameter* terbaik, model *Deep Neural Network* (DNN) dilatih ulang secara terpisah menggunakan lima variasi ukuran *batch*, yaitu 4, 16, 32, 64, dan 128. Proses pelatihan dilakukan selama 1000 *epochs* dengan menggunakan *optimizer AdamW*, dan pemilihan model terbaik pada tiap *batch size* didasarkan pada nilai *validation loss* terendah selama pelatihan. Untuk mengevaluasi pengaruh *batch size* terhadap performa model, hasil evaluasi pada data uji menggunakan metrik *Mean Squared Error* (MSE) dan *Mean Absolute Error* (MAE) dirangkum dalam Tabel 4.

Tabel 4. Evaluasi Model Berdasarkan Batch Size pada Data Uji

Batch Size	MSE	MAE
------------	-----	-----

4	0.0628	0.1992
16	0.0170	0.0676
32	0.0114	0.0519
64	0.0164	0.0738
128	0.0094	0.0490

Berdasarkan data yang tersaji pada Tabel 4, variasi *batch size* terbukti berpengaruh signifikan terhadap performa model dalam memprediksi harga saham BBKA. *Batch size* 128 menghasilkan performa terbaik dengan nilai MSE sebesar 0.0094 dan MAE sebesar 0.0490 pada data uji, menunjukkan tingkat kesalahan prediksi yang paling rendah dibandingkan *batch size* lainnya. Sebaliknya, *batch size* 4 memberikan performa terburuk dengan MSE sebesar 0.0628 dan MAE sebesar 0.1992, yang kemungkinan disebabkan oleh tingginya variabilitas antar *batch* kecil selama pelatihan, sehingga model lebih sulit melakukan generalisasi. Secara umum, *batch size* yang lebih besar cenderung menghasilkan model yang lebih stabil dengan kemampuan generalisasi yang lebih baik, meskipun penggunaan *batch size* besar memerlukan memori GPU yang lebih tinggi dan waktu pelatihan yang lebih lama. Oleh karena itu, pemilihan *batch size* harus mempertimbangkan keseimbangan antara akurasi dan efisiensi komputasi. Berdasarkan hasil evaluasi, *batch size* 128 dipilih sebagai konfigurasi terbaik untuk digunakan dalam tahap implementasi sistem prediksi harga saham BBKA.

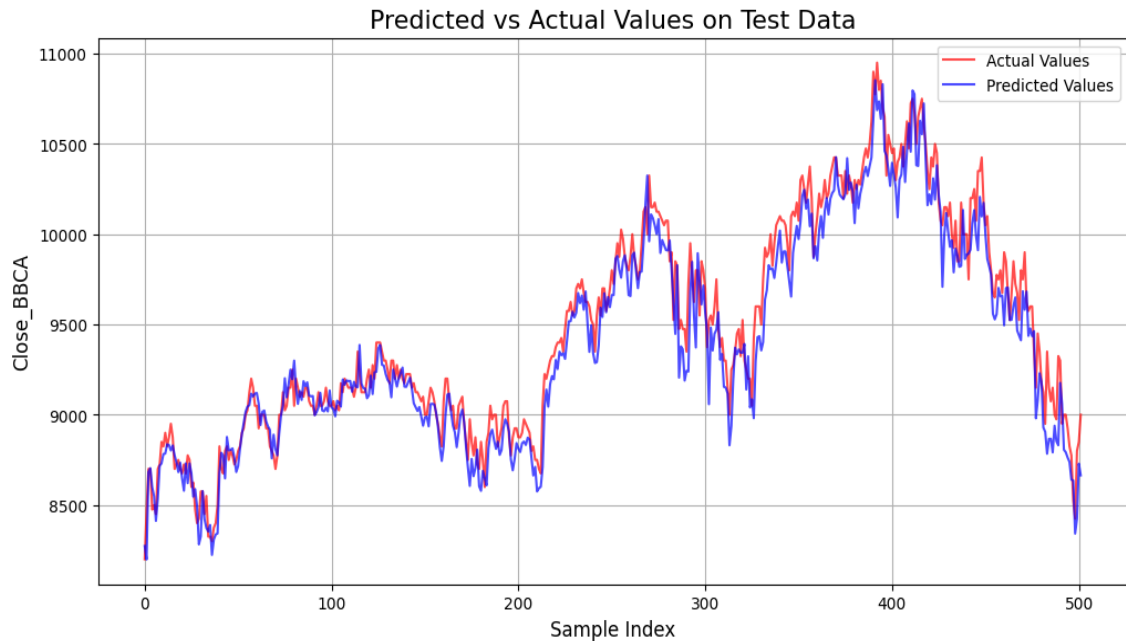
3.3 Hasil Prediksi Model dengan *Batch Size* 128 pada Data Uji

Model terbaik dengan konfigurasi *hyperparameter* optimal dan *batch size* 128 kemudian diuji menggunakan data uji untuk mengukur kemampuan generalisasi model dalam memprediksi harga penutupan saham BBKA. Evaluasi dilakukan dengan membandingkan nilai prediksi dengan nilai aktual, menggunakan metrik *Mean Squared Error* (MSE) dan *Mean Absolute Error* (MAE). Hasil evaluasi menunjukkan bahwa model menghasilkan nilai MSE sebesar 0.0094 dan MAE sebesar 0.0490, yang mengindikasikan tingkat kesalahan prediksi yang rendah. Perbandingan antara nilai aktual dan nilai prediksi dapat dilihat pada Tabel 5 berikut:

Tabel 5. Hasil Prediksi Model pada Data Uji (*Batch Size* 128)

<i>Actual (Harga Sebenarnya)</i>	<i>Predicted (Harga Prediksi)</i>
8200.0	8272.741211
8475.0	8198.871094
8700.0	8682.915039
8700.0	8704.229492
8475.0	8599.065430
8500.0	8549.125000
8450.0	8412.343750
8700.0	8533.572266
...	...
8925.0	8762.954102
8825.0	8738.804688
8775.0	8640.968750
8525.0	8635.686523
8425.0	8342.147461
8800.0	8424.853516
8850.0	8729.029297
9000.0	8664.825195

Hasil pada Tabel 5 menunjukkan bahwa nilai prediksi model sangat mendekati nilai aktual harga penutupan saham BBKA, yang mencerminkan tingkat kesalahan prediksi yang rendah. Perbedaan antara nilai aktual dan nilai prediksi relatif kecil, mengindikasikan bahwa model mampu melakukan prediksi secara akurat pada data uji. Visualisasi perbandingan antara nilai aktual dan prediksi dapat dilihat pada Gambar 2, yang menggambarkan hubungan antara hasil prediksi dengan data aktual pada data uji.



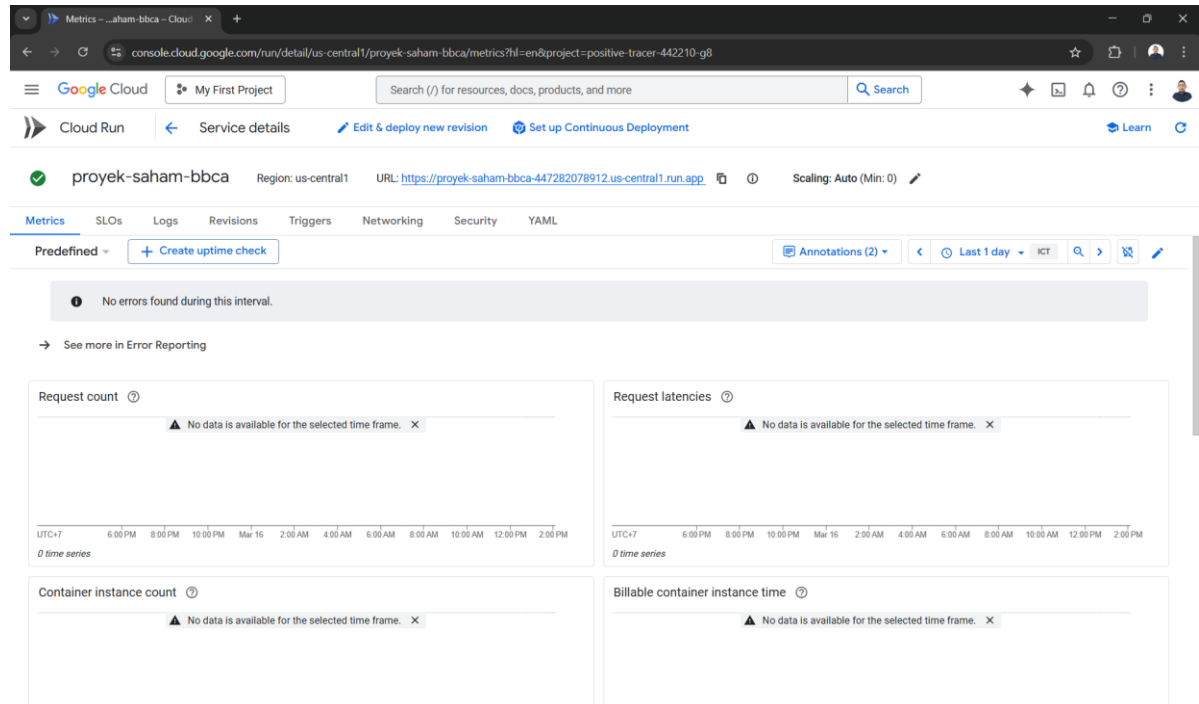
Gambar 2. Grafik Nilai Aktual dan Nilai Prediksi pada Data Uji

Gambar 2 memperlihatkan bahwa garis merah mewakili nilai aktual harga saham BBKA, sedangkan garis biru menggambarkan hasil prediksi model. Grafik ini menunjukkan bahwa pola prediksi sangat mendekati pola aktual, dengan deviasi yang kecil sepanjang rentang data uji. Sumbu horizontal (X) merepresentasikan indeks sampel data uji, yaitu urutan data berdasarkan waktu. Sumbu vertikal (Y) merepresentasikan nilai harga penutupan saham BBKA (dalam satuan rupiah). Keselarasan antara kedua garis tersebut menegaskan bahwa model DNN yang dikembangkan mampu menangkap tren dan fluktuasi harga saham secara akurat. Visualisasi ini memperkuat hasil evaluasi numerik, di mana nilai MSE dan MAE yang rendah menunjukkan performa model yang sangat baik dalam melakukan prediksi terhadap data baru.

Untuk memperkuat temuan ini, hasil penelitian dibandingkan dengan studi sebelumnya oleh Prayogi et al. (2024) yang menggunakan algoritma *Gated Recurrent Unit* (GRU) untuk memprediksi harga saham BBKA. Dalam penelitian tersebut, model GRU dikembangkan menggunakan fitur historis seperti harga pembukaan, penutupan, tertinggi, terendah, serta volume perdagangan, dengan pembagian data 80% pelatihan dan 20% pengujian. Hasil evaluasi pada data uji menunjukkan bahwa model GRU menghasilkan MSE sebesar 1.079 dan MAE sebesar 0.842 [8]. Jika dibandingkan, model DNN pada penelitian ini menunjukkan peningkatan akurasi yang signifikan, dengan nilai MSE dan MAE yang jauh lebih rendah. Pencapaian ini didukung oleh beberapa faktor, yaitu: penerapan *hyperparameter tuning* dengan *KerasTuner* dan algoritma *Hyperband*, evaluasi terhadap variasi *batch size*, serta penggunaan faktor makroekonomi seperti IHSG dan nilai tukar USD/IDR sebagai variabel *input*. Dengan performa prediksi yang tinggi dan kesalahan yang minimal, model DNN dengan *batch size* 128 dipilih sebagai model final dan digunakan pada tahap *deployment*, untuk implementasi sistem prediksi harga saham BBKA berbasis aplikasi. Hasil ini menunjukkan bahwa pendekatan DNN yang diterapkan dalam penelitian ini tidak hanya efektif, tetapi juga lebih unggul dibandingkan metode yang digunakan dalam penelitian sebelumnya.

3.4 Implementasi Model

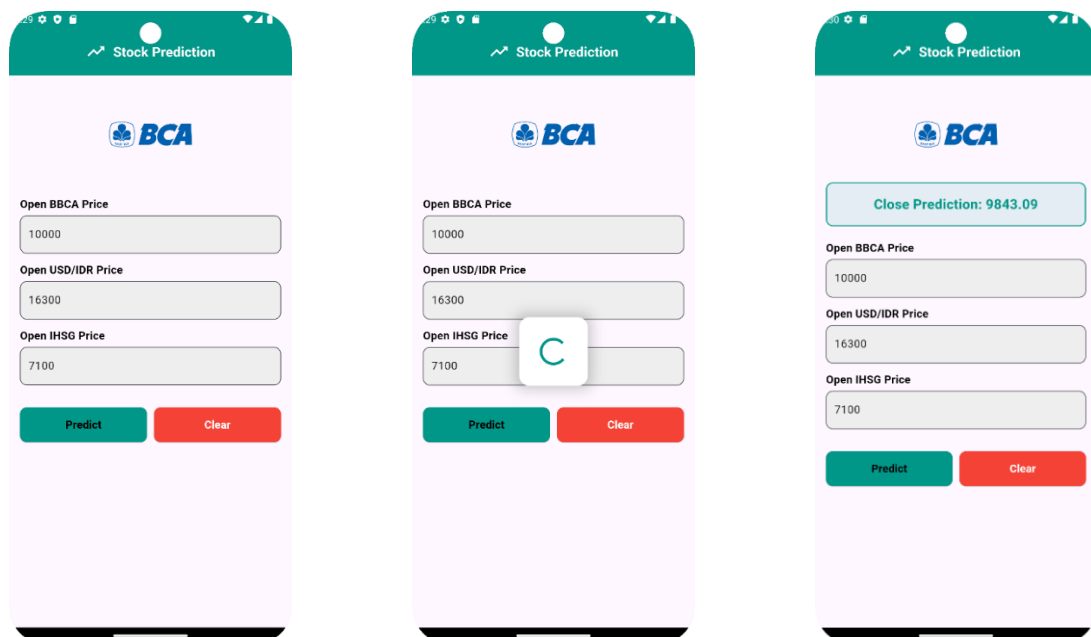
Setelah proses evaluasi model selesai dan model terbaik diperoleh dari hasil pelatihan menggunakan *batch size* 128, tahap implementasi dilakukan dengan membuat API prediksi harga saham menggunakan *framework Flask*. Model terbaik yang telah disimpan dalam format *SavedModel* digunakan sebagai inti pemrosesan prediksi pada API tersebut. API ini kemudian di-deploy ke platform *Google Cloud Platform* (GCP) menggunakan layanan *Google Cloud Run*. Proses *deployment* dilakukan dengan pendekatan berbasis *container Docker*, yang memungkinkan API berjalan secara efisien dan fleksibel dalam lingkungan *cloud*. Setelah *deployment* selesai, API diuji dengan mengirimkan data *input* dalam format JSON, yang berisi tiga parameter penting, yaitu harga pembukaan saham BBKA (*Open BBKA*), nilai tukar USD terhadap IDR (*Open USD/IDR*), dan harga pembukaan IHSG (*Open IHSG*). API yang dikembangkan berhasil mengembalikan *output* berupa prediksi harga penutupan saham BBKA (*Close BBKA*) dengan cepat dan tanpa *error*. Kondisi hasil *deployment* API ke *Cloud Run* dapat dilihat pada Gambar 3, yang menampilkan *dashboard monitoring* dari layanan yang telah diaktifkan.



Gambar 3. Hasil *Deployment API* pada *Cloud Run*

Gambar 3 menampilkan dashboard *Cloud Run* yang menunjukkan tidak adanya *error* selama interval waktu pengamatan, ditandai dengan ketiadaan laporan kesalahan (*error reporting*) yang terdeteksi. Selain itu, *panel* metrik seperti *Request Count*, *Request Latencies*, *Container Instance Count*, dan *Billable Container Instance Time* menunjukkan bahwa layanan API dalam kondisi *idle*, siap untuk menerima permintaan kapan saja. Hal ini mengindikasikan bahwa *deployment API* berjalan stabil, tidak terjadi *bottleneck*, dan sistem *cloud* berfungsi optimal untuk mendukung komunikasi antara aplikasi *mobile* dan *server API*.

Setelah API berhasil diuji, tahap selanjutnya adalah pengembangan aplikasi *mobile* berbasis *Flutter*. Aplikasi ini berfungsi sebagai antarmuka pengguna (*front-end*) yang berinteraksi langsung dengan API prediksi. Antarmuka aplikasi dirancang sederhana namun fungsional, menyediakan tiga kolom *input* untuk memasukkan nilai *Open BBKA*, *Open USD/IDR*, dan *Open IHSG*. Dua tombol utama disediakan, yaitu tombol *Predict* untuk mengirimkan permintaan prediksi ke API, serta tombol *Clear* untuk menghapus seluruh isian *form input*. Proses interaksi antara pengguna dan aplikasi *mobile* dapat dilihat secara rinci pada Gambar 4, yang memperlihatkan urutan tampilan aplikasi dari tahap *input data* hingga hasil prediksi.



Gambar 4. Tampilan Antarmuka dan Hasil Prediksi Harga Saham BBKA pada Aplikasi *Mobile*

Pada Gambar 4, bagian sebelah kiri menunjukkan tampilan awal saat pengguna mengisi kolom *input* dengan data harga pembukaan saham BBKA, nilai tukar USD/IDR, dan harga pembukaan IHSG. Bagian tengah menampilkan kondisi *loading* (indikator *loading spinner*) setelah pengguna menekan tombol *Predict* untuk mengirimkan data ke API. Bagian sebelah kanan memperlihatkan hasil prediksi harga penutupan saham BBKA (*Close Prediction*) yang ditampilkan secara *real-time* begitu respons dari API diterima. Hasil prediksi ditampilkan dalam kotak berwarna hijau berisi nilai hasil perhitungan model. Secara keseluruhan, tampilan antarmuka ini menunjukkan bahwa komunikasi antara aplikasi *mobile*, API *Flask*, dan model DNN di sisi *server* berjalan lancar tanpa hambatan, sehingga sistem dinilai layak untuk digunakan dalam skenario nyata oleh pengguna akhir.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan model *Deep Neural Network* (DNN) menggunakan *TensorFlow* untuk memprediksi harga saham PT Bank Central Asia Tbk (BBKA). Model dioptimalkan melalui *hyperparameter tuning* dengan *KerasTuner* dan diuji dengan berbagai *batch size*, di mana *batch size* 128 memberikan hasil terbaik pada data uji dengan MSE sebesar 0.0094 dan MAE sebesar 0.0490. Model ini kemudian diimplementasikan sebagai API menggunakan *Flask* dan di-deploy ke *Google Cloud Run*, serta diintegrasikan ke dalam aplikasi *mobile* berbasis *Flutter*. Sistem yang dibangun mampu memberikan prediksi harga saham secara *real-time* dengan performa yang akurat dan stabil, sehingga dapat digunakan sebagai alat bantu pengambilan keputusan investasi berbasis data.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dalam terlaksananya penelitian ini, termasuk lembaga akademik, dosen pembimbing, serta rekan-rekan yang turut berkontribusi dalam proses penelitian dan pengembangan sistem. Dukungan dan bimbingan yang diberikan sangat berperan dalam keberhasilan penelitian ini.

REFERENCES

- [1] M. F. Mahfuzh and R. V. Yuliantari, "Analisis Penerapan Artificial Neural Network Algoritma Propagasi Balik untuk Meramalkan Harga Saham pada Bursa Efek Indonesia," *J. Appl. Electr. Eng.*, vol. 6, no. 1, pp. 1–3, 2022, doi: 10.30871/jaee.v6i1.3814.
- [2] M. Imam and N. Mursidah, "Analisis Pengaruh Price Earning Ratio dan Earning Per Share terhadap Return Saham pada Sub Sektor Property and Real Estate yang Terdaftar di BEI," *Borneo Student Res.*, vol. 2, no. 2, pp. 1460–1468, 2021, [Online]. Available: <https://journals.umkt.ac.id/index.php/bsr/article/view/1959>
- [3] D. Eko Waluyo et al., "Implementasi Algoritma Regresi pada Machine Learning untuk Prediksi Indeks Harga Saham Gabungan," *Univ. Dian Nuswantoro, Semarang Jln. Imam Bonjol*, vol. 9, no. 1, pp. 12–17, 2024.
- [4] K. S. Dewi, "Pengaruh Inflasi, Suku Bunga Sbi, Dan Kurs Terhadap Indeks Harga Saham Gabungan," *J. Akuntansi, Keuangan, Pajak dan Inf.*, vol. 2, no. 2, pp. 160–180, 2022, doi: 10.32509/jakpi.v2i2.2474.
- [5] S. Soewignjo, Sediono, M. F. F. Mardianto, and E. Pusporni, "Prediksi Harga Saham Bank BCA (BBKA) Pasca Stock Split dengan Artificial Neural Network dengan Algoritma Backpropagation," *G-Tech J. Teknol. Terap.*, vol. 7, no. 4, pp. 1683–1693, 2023, doi: 10.33379/gtech.v7i4.3363.
- [6] I. G. A. D. & N. P. B. Arlita, "Stock Split Dan Return Saham (Studi Pada PT. Bank Central Asia)," *J. Innov. Res. Knowl.*, vol. 1, no. 8, pp. 689–694, 2022.
- [7] R. A. Fahrezi, M. Y. Wijaya, and N. Fitriyati, "Prediksi Harga Penutupan Saham Bank Central Asia: Implementasi Algoritma Long Short-Term Memory Dan Perbandingannya Dengan Support Vector Machine," *J. Lebesgue J. Ilm. Pendidik. Mat. Mat. dan Stat.*, vol. 5, no. 1, pp. 452–464, 2024, doi: 10.46306/lb.v5i1.582.
- [8] K. Prayogi, W. Gata, and D. P. Kussanti, "Prediksi Harga Saham Bank Central Asia Menggunakan Algoritma Deep Learning GRU," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 13, no. 1, p. 647, 2024, doi: 10.35889/jutisi.v13i1.1910.
- [9] A. Rosyd, A. Irma Purnamasari, and I. Ali, "Penerapan Metode Long Short Term Memory (Lstm) Dalam Memprediksi Harga Saham Pt Bank Central Asia," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 1, pp. 501–506, 2024, doi: 10.36040/jati.v8i1.8440.
- [10] R. Abdi, A. Rust, and T. S. Hogue, "Development of a Multilayer Deep Neural Network Model for Predicting Hourly River Water Temperature From Meteorological Data," *Front. Environ. Sci.*, vol. 9, no. September, pp. 1–13, 2021, doi: 10.3389/fenvs.2021.738322.
- [11] Y. Yu, "LSTM-Based Time Series Prediction Model : A Case Study with YFinance Stock Data," vol. 03015, pp. 1–5, 2025.
- [12] S. Molin, *Hands-On Data Analysis with Pandas*. 2021. [Online]. Available: <https://github.com/stefmolin/Hands-On-Data-Analysis-with-Pandas-2nd-edition>
- [13] G. Huang, "Missing data filling method based on linear interpolation and lightgbm," *J. Phys. Conf. Ser.*, vol. 1754, no. 1, 2021, doi: 10.1088/1742-6596/1754/1/012187.
- [14] S. P. Choy et al., "Systematic review of deep learning image analyses for the diagnosis and monitoring of skin disease," *npj Digit. Med.*, vol. 6, no. 1, pp. 1–11, 2023, doi: 10.1038/s41746-023-00914-8.
- [15] L. B. V. de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, "The choice of scaling technique matters for classification performance," *Appl. Soft Comput.*, vol. 133, pp. 1–37, 2023, doi: 10.1016/j.asoc.2022.109924.
- [16] O. A. Montesinos López, A. Montesinos López, and J. Crossa, *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. 2022. doi: 10.1007/978-3-030-89010-0.
- [17] A. R. Khan, A. T. Khan, M. Salik, and S. Bakhsh, "An Optimally Configured HP-GRU Model Using Hyperband for The Control

- of Wall Following Robot,” *Int. J. Robot. Control Syst.*, vol. 1, no. 1, pp. 66–74, 2021, doi: 10.31763/ijrcs.v1i1.281.
- [18] A. D. Jagtap and G. E. Karniadakis, “How Important Are Activation Functions in Regression and Classification? a Survey, Performance Comparison, and Future Directions,” *J. Mach. Learn. Model. Comput.*, vol. 4, no. 1, pp. 21–75, 2023, doi: 10.1615/jmachlearnmodelcomput.2023047367.
- [19] X. Jiang and C. Xu, “Deep Learning and Machine Learning with Grid Search to Predict Later Occurrence of Breast Cancer Metastasis Using Clinical Data,” *J. Clin. Med.*, vol. 11, no. 19, 2022, doi: 10.3390/jcm11195772.
- [20] Y. Kim and P. Panda, “Revisiting Batch Normalization for Training Low-Latency Deep Spiking Neural Networks From Scratch,” *Front. Neurosci.*, vol. 15, no. December, pp. 1–13, 2021, doi: 10.3389/fnins.2021.773954.
- [21] S. Khullar and N. Singh, “Water quality assessment of a river using deep learning Bi-LSTM methodology: forecasting and validation,” *Environ. Sci. Pollut. Res.*, vol. 29, no. 9, pp. 12875–12889, 2022, doi: 10.1007/s11356-021-13875-w.
- [22] P. Khatiwada and P. Dhakal, “Evaluating Serverless Machine Learning Performance on Google Cloud Run 1,” *arXiv Prepr. arXiv2406.16250*, 2024.
- [23] T. S. Gouvêa *et al.*, “Interactive Machine Learning Solutions for Acoustic Monitoring of Animal Wildlife in Biosphere Reserves,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2023-August, pp. 6405–6413, 2023, doi: 10.24963/ijcai.2023/711.
- [24] O. M. A. AL-atraqchi, “A Proposed Model for Build a Secure Restful API to Connect between Server Side and Mobile Application Using Laravel Framework with Flutter Toolkits,” *Cihan Univ. Sci. J.*, vol. 6, no. 2, pp. 28–35, 2022, doi: 10.24086/cuesj.v6n2y2022.pp28-35.