

OPTIMASI HYPERPARAMETER MODEL LSTM DAN VARIANNYA UNTUK PERAMALAN PEMBELIAN BAHAN BAKU KARET ALAM

Roy Andika*¹⁾, Kusri²⁾

1. Informatika, Universitas AMIKOM Yogyakarta, Indonesia
2. Informatika, Universitas AMIKOM Yogyakarta, Indonesia

Article Info

Kata Kunci: LSTM; Bi-LSTM; Stacked LSTM; Hyperparameter Optimization; Peramalan

Keywords: LSTM; Bi-LSTM; Stacked LSTM; Hyperparameter Optimization; Forecasting

Article history:

Received 10 March 2025

Revised 7 April 2025

Accepted 18 April 2025

Available online 1 September 2025

DOI :

<https://doi.org/10.29100/jipi.v10i3.7567>

* Corresponding author.

Corresponding Author

E-mail address:

roy7andika@gmail.com

ABSTRAK

Penelitian ini mengkaji optimasi *hyperparameter* pada model peramalan deret waktu untuk memprediksi pembelian bahan baku karet alam. Tiga arsitektur model—LSTM, Bi-LSTM, dan Stacked LSTM—dieksekusi dengan menerapkan tiga metode *tuning*, yaitu Bayesian Optimization, Hyperband, dan Optuna. Proses *tuning* dilakukan dengan mengeksplorasi berbagai kombinasi parameter, seperti jumlah *epoch*, *units*, *dropout rate*, *learning rate*, *batch size*, dan *units2*, dengan model dikompilasi menggunakan fungsi *loss* MSE dan metrik MAE. Hasil penelitian menunjukkan bahwa *learning rate* dan *dropout rate* memiliki pengaruh signifikan terhadap penurunan *error*, sedangkan peningkatan *units2* dapat meningkatkan risiko *overfitting* jika tidak diimbangi dengan strategi regularisasi yang tepat. Analisis mendalam mengungkap bahwa kombinasi Bayesian dengan Stacked LSTM menghasilkan performa terbaik pada subset data dengan *score* terendah, sedangkan Optuna menunjukkan konsistensi optimal untuk model LSTM. Menariknya, model Bi-LSTM tidak mencapai konfigurasi optimal, kemungkinan disebabkan oleh sensitivitas *tuning* yang lebih tinggi atau kompleksitas arsitektur yang tidak sesuai dengan karakteristik dataset yang digunakan. Temuan ini memberikan wawasan penting untuk pengembangan model peramalan yang lebih akurat dan efisien serta membuka peluang penelitian lanjutan dalam strategi optimasi *hyperparameter* yang adaptif dan robust.

I. PENDAHULUAN

Peramalan pembelian bahan baku merupakan aspek kritis dalam manajemen rantai pasok karena berperan dalam mengoptimalkan persediaan, mengurangi pemborosan, dan meningkatkan efisiensi produksi. Dalam industri karet alam, fluktuasi permintaan yang dipengaruhi oleh faktor ekonomi, pola musiman, dan kondisi pasar global menuntut adanya metode prediksi yang akurat agar perusahaan dapat mengambil keputusan yang tepat dalam penyediaan bahan baku [1][2][3][4][5]. Ketidaktepatan dalam peramalan tidak jarang menyebabkan kerugian finansial yang signifikan, misalnya melalui akumulasi stok berlebih (*overstock*) yang meningkatkan biaya penyimpanan maupun terjadi kekurangan stok (*stockout*) yang berdampak pada hilangnya penjualan. Berdasarkan laporan IHL Group, total biaya distorsi persediaan global—yang mencakup *overstock* dan *stockout*—diperkirakan mencapai USD 230 miliar pada periode 2020–2022 [6]. Penerapan sistem prediksi berbasis kecerdasan buatan (AI) telah menunjukkan potensi pengurangan *error* sebesar 20–50% serta penurunan biaya penyimpanan sebesar 5–10% [7]. Mengingat nilai pasar global karet alam diproyeksikan mencapai USD 18,30 miliar pada tahun 2024 [8], pengurangan biaya penyimpanan sebesar 5–10% dapat menghasilkan penghematan antara USD 915 juta hingga 1.830 juta per tahun. Data tersebut menegaskan urgensi pengembangan dan optimasi model peramalan yang tidak hanya akurat tetapi juga efisien secara komputasi, guna meminimalkan risiko finansial pada komoditas yang sangat volatil seperti karet alam.

Seiring dengan perkembangan teknologi kecerdasan buatan dan pembelajaran mesin, penerapan model *deep learning* telah banyak digunakan untuk peramalan deret waktu. Khususnya, model berbasis Long Short-Term Memory (LSTM) serta variannya, seperti Bidirectional LSTM (Bi-LSTM) dan Stacked LSTM, semakin populer karena kemampuannya menangkap pola temporal yang kompleks dan mempertahankan informasi jangka panjang, yang sering kali tidak dapat ditangani dengan baik oleh model prediktif konvensional [9][10][11].

Namun demikian, keberhasilan model LSTM sangat bergantung pada pemilihan *hyperparameter* yang optimal. Parameter seperti jumlah unit LSTM, *dropout rate*, *batch size*, dan *learning rate* memiliki dampak signifikan terhadap kinerja model. Pengaturan *hyperparameter* yang tidak tepat dapat menyebabkan terjadinya *overfitting*—di mana model terlalu menyesuaikan diri dengan data latih sehingga tidak mampu menggeneralisasi data baru—atau *underfitting*, yang mengakibatkan prediksi menjadi kurang akurat [12]. Oleh karena itu, diperlukan metode

optimasi yang efisien untuk menentukan kombinasi *hyperparameter* terbaik dalam rangka meningkatkan performa model.

Optimasi *hyperparameter* merupakan tantangan tersendiri dalam implementasi model *deep learning* karena melibatkan banyaknya kemungkinan kombinasi parameter yang harus dievaluasi. Pendekatan konvensional seperti *grid search* dan *random search* memiliki keterbatasan efisiensi, mengingat metode tersebut memerlukan waktu komputasi yang cukup besar untuk mengeksplorasi ruang parameter [12]. Sebagai solusinya, telah dikembangkan berbagai metode optimasi *hyperparameter* yang lebih canggih, antara lain Bayesian Optimization, Hyperband, dan Optuna. Bayesian Optimization merupakan metode yang efisien untuk mengoptimasi fungsi objektif yang mahal dievaluasi (*expensive black-box functions*). Metode ini menggunakan model probabilistik—misalnya Proses Gaussian—untuk memprediksi kinerja fungsi kemudian memilih titik evaluasi berikutnya dengan memanfaatkan fungsi akuisisi yang menyeimbangkan antara eksplorasi dan eksploitasi, sehingga jumlah evaluasi langsung dapat diminimalkan [13]. Sementara itu, Hyperband merupakan algoritma yang berbasis pada alokasi sumber daya adaptif, mengombinasikan pencarian acak dengan teknik *successive halving*. Dengan menjalankan eksperimen secara paralel dan memberlakukan mekanisme pemberhentian dini (*early stopping*) bagi konfigurasi yang berkinerja buruk, Hyperband mampu mengalihkan sumber daya kepada kandidat terbaik secara efisien, terutama untuk skenario dengan banyaknya konfigurasi *hyperparameter*. Kedua metode tersebut menekankan efisiensi dengan pendekatan yang berbeda, di mana Bayesian Optimization berfokus pada pendekatan probabilistik dan Hyperband mengedepankan kecepatan serta eliminasi dinamis [14]. Selain itu, framework Optuna menghadirkan fleksibilitas dengan fitur *pruning* berbasis *Tree-structured Parzen Estimator* (TPE) yang secara adaptif menghentikan eksperimen yang tidak menjanjikan, sehingga dapat menghemat waktu komputasi tanpa mengorbankan kualitas optimasi [13][15].

Karakteristik peramalan pada komoditas seperti karet alam juga menuntut model untuk mampu menangkap dinamika temporal yang beragam, mulai dari fluktuasi harga cepat, pola musiman yang kompleks, hingga ketidakstabilan jangka pendek. Varian LSTM—termasuk Bi-LSTM dan Stacked LSTM—dirancang khusus untuk memproses data berurutan melalui mekanisme *gated recurrence*. Mekanisme tersebut memungkinkan model untuk secara selektif mempertahankan informasi penting dan menghilangkan informasi yang kurang relevan, sekaligus mengatasi masalah *vanishing gradient* yang umum terjadi pada RNN konvensional [16]. Selain itu, LSTM memiliki keunggulan dalam hal efisiensi komputasi, karena dapat dilatih secara sekuensial tanpa memerlukan ukuran *batch* yang besar atau arsitektur paralel yang kompleks. Sebagai perbandingan, arsitektur Transformer yang mengandalkan mekanisme *self-attention* dan *positional encoding* memerlukan *volume data* yang besar dan kapasitas komputasi tinggi, serta terkadang kurang responsif terhadap perubahan pola musiman serta volatilitas jangka pendek [17]. Model hibrida CNN-LSTM, meskipun menawarkan keuntungan dalam mengekstraksi fitur lokal melalui lapisan konvolusional, cenderung meningkatkan kompleksitas arsitektur, risiko *overfitting*, serta menurunkan interpretabilitas model [16]. Oleh karena itu, keunggulan LSTM menjadikannya pilihan yang tepat untuk diterapkan dalam penelitian ini.

Dengan demikian, varian LSTM dipilih sebagai model dalam penelitian ini karena kemampuannya yang unggul dalam menangani berbagai aspek penting. Pertama, LSTM mampu menangkap dinamika temporal multiskala, mulai dari pola jangka pendek hingga musiman, melalui mekanisme *cell state* dan *forget gate* yang dimilikinya. Kedua, model ini dapat beradaptasi dengan baik pada dataset yang terbatas karena jumlah parameternya lebih terkendali dibandingkan dengan model Transformer, sekaligus menawarkan kemampuan generalisasi yang baik. Selain itu, LSTM juga menjaga efisiensi komputasi pada infrastruktur penelitian yang tersedia, tanpa mengorbankan akurasi prediksi, bahkan pada *time series* yang sangat fluktuatif. Kombinasi keunggulan ini menjadikan LSTM sebagai pilihan yang tepat untuk penelitian ini.

Penelitian ini akan menganalisis secara mendalam pengaruh metode optimasi *hyperparameter* terhadap performa model LSTM, Bi-LSTM, dan Stacked LSTM dalam peramalan pembelian bahan baku karet alam. Dengan membandingkan efektivitas tiga metode optimasi—yaitu Bayesian Optimization, Hyperband, dan Optuna—tujuan penelitian ini adalah untuk menentukan metode optimasi yang paling efisien dan akurat dalam meningkatkan kinerja model prediksi. Hasil penelitian diharapkan dapat memberikan strategi optimasi *hyperparameter* yang tidak hanya bermanfaat bagi industri karet alam, tetapi juga aplikatif untuk peramalan deret waktu dalam bidang lain yang berbasis *deep learning*.

Penelitian [12] mengeksplorasi optimasi *hyperparameter* untuk algoritma *machine learning* dalam konteks analisis sentimen berbahasa Arab. Studi ini membandingkan lima metode *tuning*: Grid Search, Random Search, Bayesian Optimization, Particle Swarm Optimization (PSO), dan Genetic Algorithm (GA), dengan menerapkannya pada enam model klasifikasi (Logistic Regression, Ridge Classifier, SVM, Decision Tree, Random Forest, dan Naive Bayes). Dataset yang digunakan terdiri dari 7.000 ulasan hotel berbahasa Arab, dan hasilnya menunjukkan bahwa

Bayesian Optimization mampu mencapai akurasi tertinggi (95,62%) pada model Support Vector Classifier (SVC). Meskipun demikian, penelitian ini mengidentifikasi beberapa keterbatasan, seperti belum diujicobanya metode optimasi modern seperti Hyperband atau Optuna yang berbasis *early-stopping* atau *pruning*. Selain itu, fokus penelitian terbatas pada algoritma *machine learning* konvensional, sementara model *deep learning* seperti LSTM, Bi-LSTM, atau Stacked LSTM—yang memiliki kompleksitas *hyperparameter* lebih tinggi—belum dieksplorasi. Hal ini membuka peluang bagi penelitian lanjutan untuk menguji efektivitas metode optimasi tersebut pada arsitektur *deep learning*, terutama dalam menangani tantangan bahasa Arab yang morfologis kompleks.

Penelitian [13] berfokus pada optimasi *hyperparameter* model *deep learning* untuk prediksi daya angin, yang merupakan tantangan penting dalam integrasi energi terbarukan ke dalam jaringan listrik. Meskipun berbagai metode telah dikembangkan untuk memprediksi daya angin, penelitian ini menyoroti bahwa banyak pendekatan tradisional, seperti *grid search* dan *random search*, sering kali tidak efisien dan tidak dapat diandalkan, terutama untuk model yang kompleks seperti *deep neural network*. Dalam studi ini, peneliti membandingkan tiga teknik optimasi *hyperparameter* canggih—Scikit-opt, Optuna, dan Hyperopt—dalam konteks model Convolutional Neural Network (CNN) dan LSTM. Hasil menunjukkan bahwa algoritma Optuna dengan metode pencarian *Tree-structured Parzen Estimator* (TPE) dan fungsi akuisisi *Expected Improvement* (EI) memberikan efisiensi terbaik untuk kedua model. Namun, meskipun penelitian ini memberikan wawasan berharga tentang pengaruh teknik optimasi yang berbeda, masih ada kekurangan yang perlu ditangani dalam penelitian selanjutnya. Salah satunya adalah kurangnya eksplorasi terhadap dampak dari variasi acak dalam inisialisasi bobot pada kinerja model, yang dapat mempengaruhi stabilitas dan keandalan prediksi. Penelitian mendatang dapat memperluas analisis ini dengan menguji lebih banyak variasi dalam inisialisasi dan mengembangkan metode yang lebih robust untuk mengatasi ketidakpastian yang dihasilkan dari faktor acak, sehingga meningkatkan akurasi dan konsistensi model prediksi daya angin.

Penelitian [18] mengeksplorasi teknik optimasi *hyperparameter* dalam sistem AutoML, dengan fokus pada penggunaan algoritma evolusi untuk meningkatkan kinerja model *machine learning* dalam klasifikasi citra. Peneliti membandingkan metode seperti *grid search*, Bayesian optimization (BO), dan algoritma genetik, serta menguji kombinasi BO dengan Differential Evolution (DE) dan Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Hasil menunjukkan bahwa kombinasi BO dengan DE dan CMA-ES lebih efektif dibandingkan BO konvensional, meskipun algoritma genetik kurang berhasil. Namun, penelitian ini memiliki kekurangan, yaitu kurangnya eksplorasi variasi parameter dalam algoritma evolusi yang dapat mempengaruhi hasil. Penelitian selanjutnya dapat memperdalam analisis ini dengan menguji variasi parameter dan teknik optimasi hibrida yang lebih kompleks, sehingga dapat memberikan pemahaman yang lebih mendalam tentang optimasi *hyperparameter* di berbagai konteks, termasuk dalam peramalan berbasis LSTM.

Berbagai studi telah berhasil mengembangkan teknik optimasi *hyperparameter* yang meningkatkan akurasi model, namun sebagian besar belum menyelesaikan *trade-off* krusial antara skalabilitas komputasi dan presisi prediksi—terutama dalam konteks komoditas volatil seperti karet alam. Misalnya, metode Bayesian Optimization konvensional yang dipercepat melalui Fabolas terbukti mampu mengurangi waktu pencarian *hyperparameter* pada dataset besar, namun masih memerlukan pengambilan subset data yang agresif sehingga berpotensi menurunkan akurasi prediksi pada *time series* dengan fluktuasi tinggi [19]. Pendekatan *weighted error function* seperti GridSearchWEF yang diusulkan oleh Zhao et al. berhasil menyeimbangkan biaya komputasi dan performa pada *forecasting* minyak mentah, tetapi belum dievaluasi pada skala dan karakteristik volatilitas data karet alam [20]. Upaya lain, seperti CARBS—*Cost-Aware Pareto Region Bayesian Search*—memperkenalkan optimasi yang adaptif terhadap beban komputasi dan akurasi, namun penerapannya pada domain peramalan komoditas volatil masih sangat terbatas [21]. Sementara itu, RASDA menunjukkan skalabilitas luar biasa pada infrastruktur HPC dengan ribuan GPU untuk dataset ilmiah berukuran *terabyte*, tetapi fokusnya belum menyentuh tantangan volatilitas pasar komoditas [22]. Oleh karena itu, penelitian ini berupaya menjembatani kesenjangan tersebut dengan membandingkan tiga metode optimasi *hyperparameter* yang tidak hanya efisien secara komputasi tetapi juga *robust* terhadap karakteristik data karet alam yang sangat fluktuatif.

Sebagian besar penelitian hingga saat ini menerapkan metode optimasi *hyperparameter* secara terpisah, tanpa mengeksplorasi sinergi antara pendekatan cepat seperti Hyperband dan teknik pencarian lokal yang presisi seperti Bayesian Optimization. Hyperband dapat digunakan sebagai tahap *initial screening* untuk secara efisien mengevaluasi sejumlah besar konfigurasi *hyperparameter* dengan anggaran komputasi terbatas, sedangkan Bayesian Optimization selanjutnya mengoptimalkan subset konfigurasi terpilih untuk mencapai akurasi prediksi yang lebih tinggi [13][23]. Pendekatan hibrida ini—sering disebut BOHB (*Bayesian Optimization Hyperband*)—telah terbukti unggul dalam domain seperti peramalan energi angin dan *general time series*, namun aplikasinya pada model LSTM untuk data komoditas volatil seperti karet alam masih sangat terbatas. Dengan menggabungkan kekuatan eksplorasi Hyperband dan eksploitasi Bayesian Optimization, strategi ini berpotensi meningkatkan efisiensi

komputasi sekaligus mempertahankan atau bahkan meningkatkan akurasi prediksi pada dataset yang memiliki fluktuasi tinggi.

Meskipun *framework hybrid* seperti BOHB telah diusulkan untuk menggabungkan kecepatan Hyperband dan presisi Bayesian Optimization dalam satu pipeline—misalnya melalui *multi-objective* BOHB dengan *weighted metric scalarization* [24] atau penerapan BOHB pada peramalan harga listrik harian [25]—penelitian ini memilih pendekatan komparatif terpisah dengan membandingkan tiga metode optimasi *hyperparameter*—Bayesian Optimization, Hyperband, dan Optuna—pada model LSTM, Bi-LSTM, dan Stacked LSTM tanpa mengintegrasikan *framework hybrid*. Strategi ini memungkinkan analisis mendalam atas kelebihan dan kekurangan masing-masing metode dalam konteks peramalan komoditas volatil seperti karet alam, serta memberikan rekomendasi yang lebih langsung bagi praktisi yang bekerja dengan batasan komputasi dan skala dataset nyata [14].

II. METODE PENELITIAN

A. Data Penelitian

Penelitian ini menggunakan dua dataset utama untuk mendukung proses pemodelan. Dataset pertama berisi rekaman pembelian bahan baku karet alam dari perusahaan pengolahan karet remah (*crumb rubber*) di Kalimantan selama empat tahun terakhir (2021-2024). Data ini mencakup variabel seperti tanggal pembelian, jumlah bahan baku, harga total, nilai USD, nilai TSR20, hari kalender, serta kondisi cuaca harian (rerata kelembaban dan curah hujan). Sebelum digunakan untuk pelatihan model, dataset ini melalui tahapan *preprocessing*:

- 1) *Encoding* Kategorikal menggunakan LabelEncoder untuk konversi variabel kategorikal ke numerik,
- 2) Normalisasi Data dengan Robust Scaler untuk meminimalkan pengaruh *outlier*,
- 3) Pembagian Dataset dengan rasio 80% data latih, 10% validasi, dan 10% uji guna memastikan evaluasi model yang objektif.

Setelah *preprocessing*, dilakukan pelatihan model LSTM dan variannya (Bi-LSTM dan Stacked LSTM) dengan menjalankan tiga metode optimasi *hyperparameter* secara paralel: Bayesian Optimization, Hyperband, dan Optuna. Setiap metode diuji pada konfigurasi *epoch* yang berbeda untuk mengeksplorasi kombinasi *hyperparameter* optimal seperti jumlah *unit* neuron, tingkat *dropout*, *learning rate*, dan fungsi aktivasi.

Hasil eksperimen tuning ini kemudian dikompilasi menjadi Dataset kedua yang berisi 1.298 baris data dengan 11 kolom, termasuk detail kombinasi *hyperparameter*, metode tuning, dan skor performa model berdasarkan evaluasi MSE dan MAE. Dataset ini tidak hanya merekam parameter teknis seperti *batch_size*, *activation function*, dan *optimizer*, tetapi juga menyertakan kolom khusus *units2* untuk konfigurasi lapisan kedua pada model Stacked LSTM. Skor performa (*score*) dalam dataset ini menjadi acuan utama dalam menentukan arsitektur model akhir yang paling optimal untuk prediksi pembelian bahan baku karet.

B. Arsitektur Model

Penelitian ini membandingkan tiga jenis model berbasis LSTM untuk mengoptimalkan performa peramalan:

- 1) LSTM (*Long Short-Term Memory*)
Model dasar yang menangani peramalan dengan mempertahankan informasi jangka panjang dalam urutan data.
- 2) Bi-LSTM (*Bidirectional LSTM*)
Model yang mengolah data dalam dua arah (maju dan mundur) untuk menangkap informasi kontekstual yang lebih kaya.
- 3) Stacked LSTM
Model yang terdiri dari beberapa lapisan LSTM bertingkat untuk menangkap fitur temporal yang lebih kompleks.

Konfigurasi umum model yang digunakan dalam penelitian ini adalah sebagai berikut:

- *Loss function*: Mean Squared Error (MSE)
- *Metrics*: Mean Absolute Error (MAE)
- *Epoch*: 50, 100, 200
- *Early Stopping*: Diterapkan dengan *patience* = 10 untuk menghentikan pelatihan jika performa validasi tidak membaik selama 10 *epoch* berturut-turut.

C. Metode Tuning (Optimasi Hyperparameter)

Untuk mendapatkan kombinasi *hyperparameter* terbaik, penelitian ini menggunakan tiga metode *tuning* yang berbeda:

- 1) Bayesian Optimization

Metode ini menggunakan pendekatan probabilistik untuk memodelkan fungsi objektif secara iteratif dan memilih kombinasi hyperparameter berdasarkan akuisisi fungsi terbaik.

2) Hyperband

Hyperband adalah metode tuning berbasis *successive halving* yang secara adaptif mengalokasikan sumber daya pelatihan ke kombinasi hyperparameter yang menjanjikan.

3) Optuna

Optuna adalah *framework* optimisasi *hyperparameter* berbasis *sampling* adaptif yang menggunakan algoritma *Tree-structured Parzen Estimator* (TPE) untuk eksplorasi ruang parameter.

Parameter yang dioptimalkan dalam proses *tuning* ini meliputi:

- Jumlah unit *neuron* (units dan units2): 16, 32, 48, 64, 80, 96, 112, 128
- *Dropout rate*: 0.0, 0.1, 0.2, 0.3, 0.4, 0.5
- *Learning rate*: 0.0001 hingga 0.01
- *Activation*: ReLU, Tanh, Sigmoid
- *Optimizer*: Adam, Nadam, RMSprop, SGD
- *Batch size*: 16, 32, 64, 128

Setiap kombinasi *hyperparameter* diuji dengan masing-masing metode *tuning*, dan hasilnya dibandingkan berdasarkan performa model.

D. Evaluasi Kinerja Model

Performa model dinilai berdasarkan metrik evaluasi *Mean Squared Error* (MSE) dan *Mean Absolute Error* (MAE) pada data validasi dan uji. Model terbaik ditentukan berdasarkan kombinasi *hyperparameter* yang menghasilkan nilai MSE dan MAE paling rendah.

MAE adalah salah satu metrik yang digunakan untuk mengevaluasi kinerja model prediksi dengan mengukur rata-rata dari nilai absolut kesalahan antara nilai yang diprediksi dan nilai yang sebenarnya. MAE dihitung dengan menjumlahkan semua kesalahan absolut (selisih antara nilai yang diprediksi dan nilai yang sebenarnya) dan membaginya dengan jumlah observasi. Metrik ini memberikan gambaran yang jelas tentang seberapa besar kesalahan yang terjadi dalam prediksi, tanpa memberikan bobot lebih pada kesalahan yang lebih besar, seperti yang dilakukan oleh RMSE [26]. Rumus dari MAE disajikan sebagai persamaan (1) berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1)$$

Keterangan:

n = jumlah data

y_i = nilai aktual pada data ke- i

\hat{y}_i = nilai prediksi pada data ke- i

$|y_i - \hat{y}_i|$ = selisih absolut antara nilai aktual dan nilai prediksi

MAE sangat berguna dalam situasi di mana kesalahan besar tidak diinginkan, dan lebih baik untuk memiliki metrik yang memberikan representasi yang lebih stabil dari kesalahan rata-rata. Metrik ini optimal untuk kesalahan yang terdistribusi Laplace, di mana kesalahan cenderung memiliki ekor yang lebih berat dibandingkan dengan distribusi normal [26].

MSE adalah sebuah metrik yang digunakan untuk menilai akurasi model regresi dengan menghitung rata-rata dari kuadrat selisih antara nilai yang diprediksi dan nilai aktual. Metrik ini memberikan penekanan lebih pada kesalahan yang lebih besar, karena kesalahan dihitung dengan cara mengkuadratkan selisihnya. Dengan demikian, MSE sangat sensitif terhadap *outlier*, yang berarti bahwa satu prediksi yang sangat buruk dapat secara signifikan meningkatkan nilai MSE. Nilai MSE yang lebih rendah menunjukkan bahwa model memiliki kinerja yang lebih baik, dengan nilai terbaik adalah nol, yang menunjukkan bahwa tidak ada kesalahan dalam prediksi [27]. Rumus dari MSE disajikan sebagai persamaan (2) berikut:

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

Keterangan:

n = jumlah data

y_i = nilai aktual pada data ke- i

\hat{y}_i = nilai prediksi pada data ke- i

$(y_i - \hat{y}_i) = \text{selisih absolut antara nilai aktual dan nilai prediksi}$

E. Pelatihan Model

Proses pelatihan model dimulai dengan pembagian dataset menjadi tiga bagian: 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Data kemudian diproses melalui tahap *preprocessing*, yang mencakup *encoding* variabel kategorikal dengan LabelEncoder dan normalisasi fitur numerik menggunakan Robust Scaler, sehingga data siap untuk digunakan dalam pelatihan. Selanjutnya, dilakukan *tuning hyperparameter* dengan mengeksplorasi berbagai kombinasi parameter menggunakan tiga metode *tuning*, yaitu Bayesian Optimization, Hyperband, dan Optuna, digunakan untuk mencari konfigurasi optimal. Model dikompilasi menggunakan fungsi *loss* MSE dan dilengkapi dengan metrik MAE untuk monitoring performa selama pelatihan. Proses pelatihan dilakukan dengan beberapa nilai *epoch* (50, 100, dan 200) dan dilengkapi dengan mekanisme *Early Stopping* (*patience*=10) guna mencegah *overfitting*. Setiap kombinasi *hyperparameter* dievaluasi pada data validasi untuk menentukan konfigurasi terbaik yang kemudian diuji pada data pengujian guna mengukur performa akhir model.

III. HASIL PENELITIAN

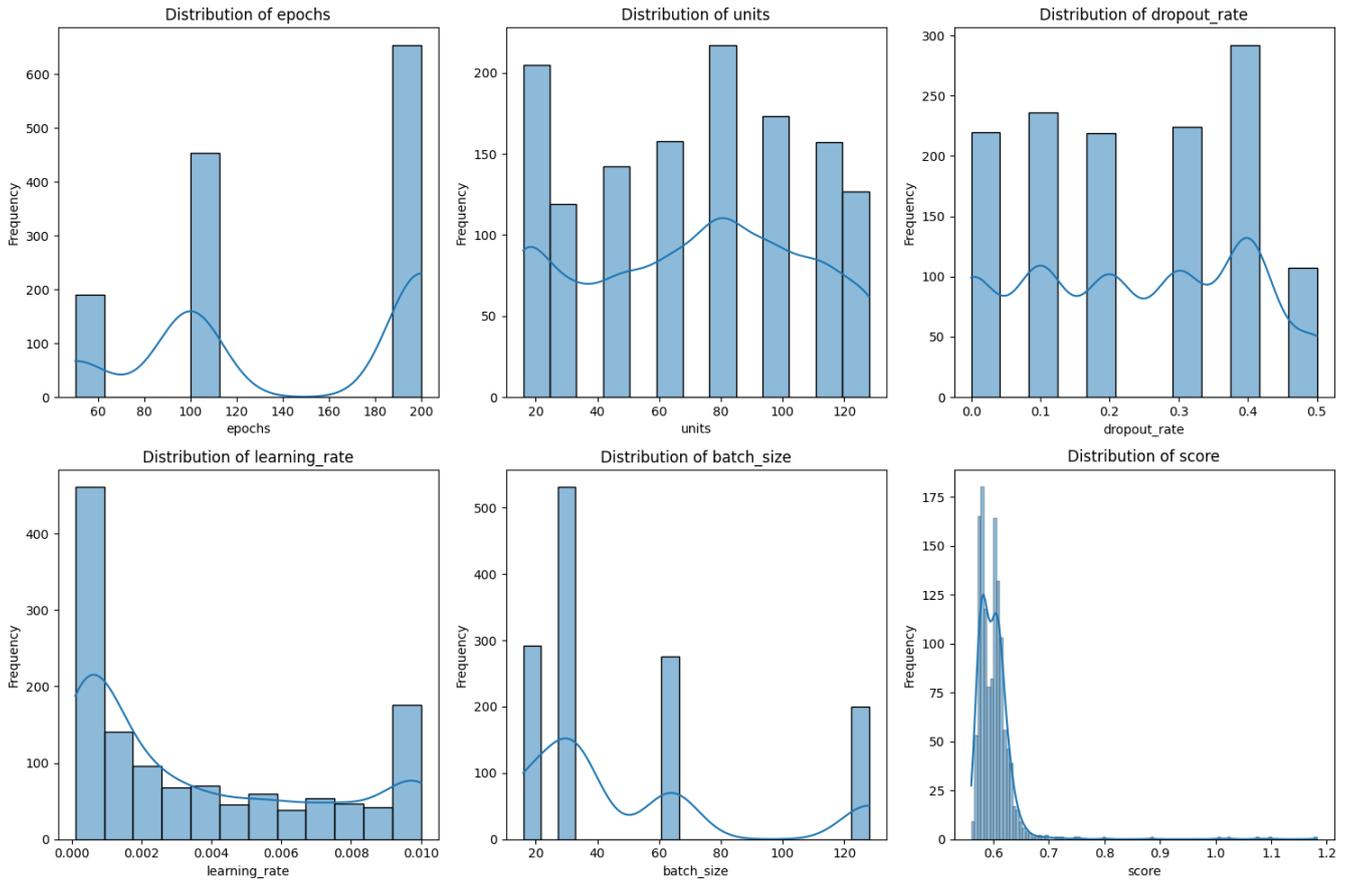
A. Eksplorasi Dataset Hasil Pengujian

Dataset yang dianalisis terdiri dari 1.298 baris data dengan 11 kolom, di mana 4 kolom di antaranya bersifat kategorikal dan 7 kolom numerik. Kolom kategorikal mencakup *tuning_method*, *model_type*, *activation*, dan *optimizer*, sementara kolom numerik meliputi *epochs*, *units*, *dropout_rate*, *learning_rate*, *batch_size*, *units2*, dan *score*.

Dari segi statistik dasar, variasi parameter terlihat jelas. Jumlah *epochs* ditentukan secara manual pada tiga angka tetap, yaitu 50, 100, dan 200, selama proses *tuning* dijalankan. Sementara itu, jumlah *units* rata-rata adalah 70, meskipun nilainya bervariasi dari 16 hingga 128. *Dropout rate* umumnya berada di rentang 0 hingga 0.5, sedangkan *learning rate* cenderung rendah, yaitu antara 0.0001 hingga 0.01. Untuk *batch size*, nilai mediannya adalah 32, meskipun distribusinya cukup merata dari 16 hingga 128. Performa model (*score*) menunjukkan rata-rata 0.60, dengan rentang nilai 0.56 hingga 1.18.

Terdapat ketidaklengkapan data pada kolom *units2*, di mana 301 dari 1.298 data bernilai kosong. Hal ini disebabkan karena kolom *units2* hanya relevan untuk model dengan tipe Stacked LSTM, yang menggunakan arsitektur dua lapisan. Sementara itu, model LSTM dan Bi-LSTM (Bidirectional LSTM) tidak memerlukan parameter ini, karena hanya menggunakan satu lapisan atau pendekatan dua arah tanpa penumpukan lapisan tambahan.

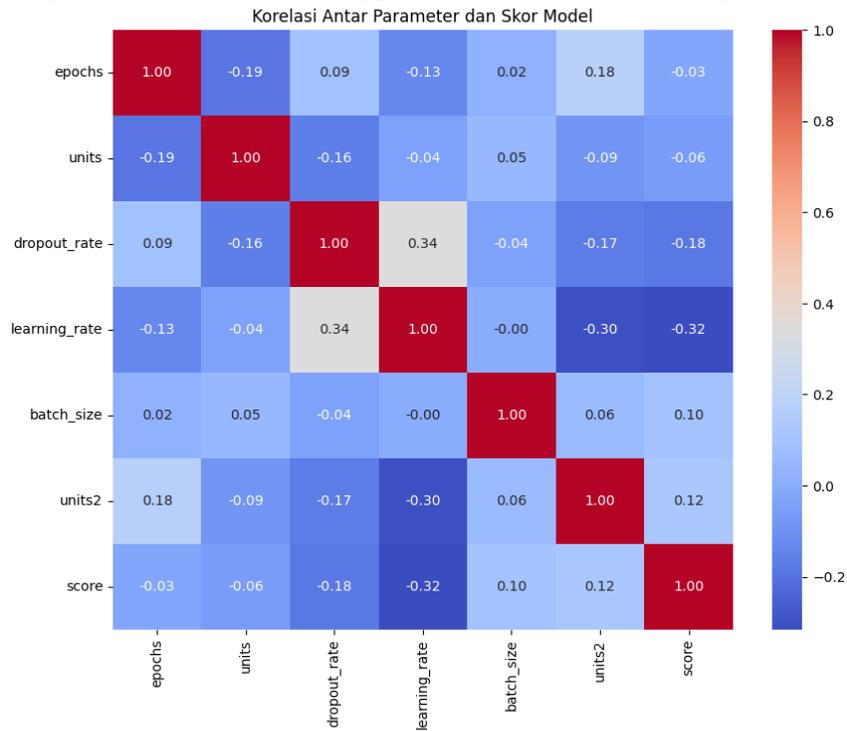
Analisis visual melalui grafik distribusi *hyperparameter* pada Gambar 1, mengungkap beberapa pola menarik. *Epochs* lebih sering diatur pada 100 atau 200, menunjukkan preferensi terhadap pelatihan yang lebih lama. Distribusi *units* relatif merata, tetapi terdapat lonjakan pada nilai 20 dan 80, yang mungkin merefleksikan konfigurasi umum dalam optimasi model. *Dropout rate* banyak terkonsentrasi di kisaran 0.1–0.3 dan terjadi lonjakan pada 0.4, mengindikasikan upaya mitigasi *overfitting* tanpa mengurangi kapasitas model secara drastis. *Learning rate* didominasi nilai di bawah 0.001, menegaskan preferensi terhadap pembelajaran bertahap. *Batch size* paling sering menggunakan 32, dengan distribusi yang cenderung merata pada 20, 64 dan 128, menunjukkan keberagaman distribusi. Untuk performa model, sebagian besar *score* berada di rentang 0.58–0.62, dengan beberapa *outlier* di atas 1.0 yang mungkin mencerminkan kegagalan atau ketidakstabilan pelatihan.



Gambar 1. Distribusi Hyperparameter

B. Analisis Korelasi Hyperparameter

Analisis korelasi *hyperparameter* dilakukan menggunakan visualisasi berupa seperti Gambar 2 berikut:



Gambar 2. Diagram Heatmap Korelasi Hyperparameter

TABEL I.
 INTERPRETASI KORELASI HYPERPARAMETER BERDASARKAN DIAGRAM HEATMAP

Hyperparameter	Korelasi terhadap score	Interpretasi
<i>learning_rate</i>	-0.317	Pengaruh terkuat: Penurunan learning rate berkorelasi dengan skor lebih rendah, menunjukkan model lebih baik.
<i>dropout_rate</i>	-0.181	Tingkat <i>dropout</i> yang lebih tinggi meningkatkan generalisasi model dan mengurangi risiko <i>overfitting</i> .
<i>units2</i>	+0.122	Penambahan unit di lapisan kedua cenderung menurunkan performa, kemungkinan karena peningkatan risiko <i>overfitting</i> .
<i>batch_size</i>	+0.096	<i>Batch size</i> yang besar sedikit berkorelasi dengan skor yang lebih buruk, yang mungkin mengurangi stabilitas pelatihan.
<i>epochs</i>	-0.028	Jumlah <i>epoch</i> tidak memberikan pengaruh signifikan terhadap skor, menandakan bahwa pelatihan mungkin sudah mencapai konvergensi.

Tabel 1 merangkum korelasi antar *hyperparameter* dengan *score* beserta interpretasinya. Analisis dari tabel tersebut mengungkapkan bahwa *learning_rate* memiliki pengaruh yang paling dominan terhadap performa model. Penurunan *learning_rate* secara konsisten menghasilkan nilai *score* yang lebih rendah, yang mengindikasikan bahwa model belajar dengan lebih stabil dan akurat. *dropout_rate* juga memiliki peran penting; nilai *dropout* yang lebih tinggi berkontribusi dalam meningkatkan generalisasi model dan mengurangi *overfitting*, meskipun pengaruhnya tidak sebesar *learning_rate*. Sebaliknya, peningkatan jumlah unit di lapisan kedua (*units2*) menunjukkan korelasi positif dengan skor, yang berarti bahwa menambah terlalu banyak neuron pada lapisan tersebut dapat menimbulkan kompleksitas berlebih dan berpotensi meningkatkan error. *Batch_size* yang lebih besar juga memiliki korelasi positif, menunjukkan bahwa penggunaan *batch* yang terlalu besar bisa mengganggu stabilitas pelatihan. Sedangkan jumlah *epoch*, dengan korelasi yang sangat kecil, tampaknya tidak terlalu mempengaruhi performa, mengindikasikan bahwa pelatihan sudah mencapai titik konvergensi pada nilai *epoch* yang digunakan.

Secara keseluruhan, temuan ini menekankan pentingnya penyesuaian *learning_rate* dan *dropout_rate* sebagai kunci utama dalam optimasi model, sementara penambahan kompleksitas melalui *units2* harus dilakukan secara hati-hati untuk menghindari *overfitting*.

Meskipun analisis korelasi linear memberikan gambaran awal mengenai hubungan antar *hyperparameter*, pendekatan tersebut memiliki keterbatasan dalam mendeteksi interaksi non-linear yang kompleks. Interaksi variabel seperti *learning_rate*, *dropout_rate*, dan jumlah unit mungkin membentuk pola sinergis yang tidak dapat dijelaskan hanya dengan koefisien linear. Oleh karena itu, eksplorasi hubungan non-linear—misalnya melalui regresi non-linear atau teknik visualisasi multivariat—diperlukan untuk mengungkap kompleksitas interaksi antar parameter. Pendekatan ini dapat membuka wawasan baru terkait kombinasi *hyperparameter* yang secara potensial menghasilkan kondisi optimal dalam pelatihan model, terutama dalam konteks data karet alam yang memiliki dinamika temporal yang kompleks.

C. Performa Model Berdasarkan Tipe Model

Untuk memahami performa tiap tipe model, dilakukan analisis distribusi *score* menggunakan visualisasi *boxplot* untuk model pada gambar 3. Visualisasi menunjukkan bahwa model LSTM dan Bi-LSTM memiliki variasi *score* yang lebih tinggi, sedangkan Stacked LSTM cenderung lebih stabil, dengan nilai *score* yang lebih rendah pada beberapa kombinasi *hyperparameter*.

Hasil *tuning hyperparameter* menggunakan tiga metode berbeda menghasilkan kombinasi terbaik yang memiliki karakteristik unik dan memberikan wawasan mendalam tentang pengaruh tipe model, konfigurasi *hyperparameter*, dan metode *tuning* terhadap performa model.

1) Stacked LSTM

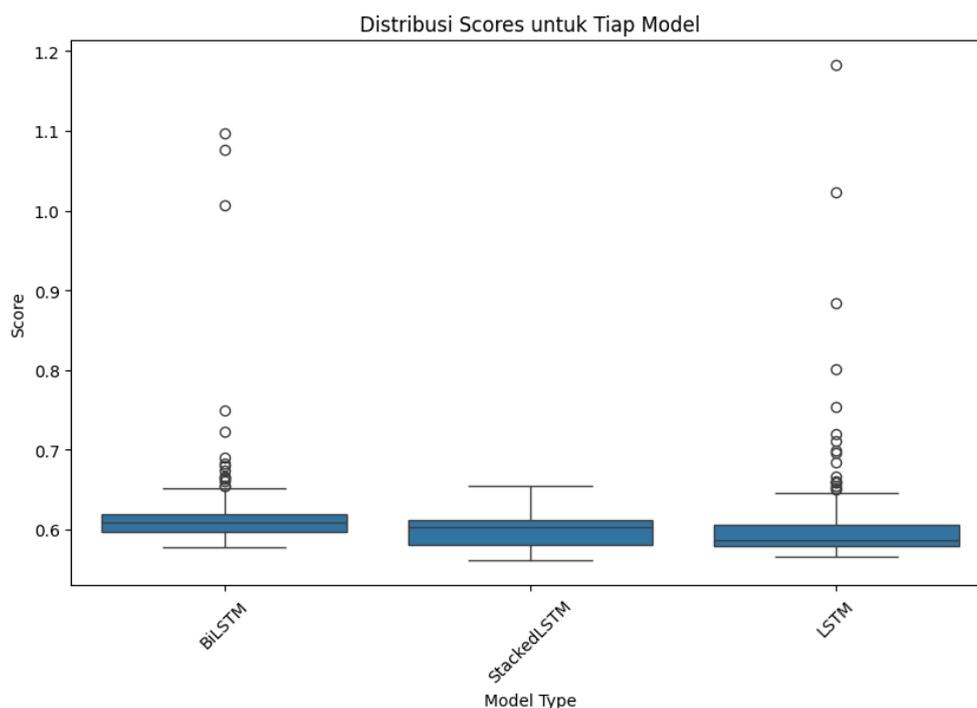
Score 0.560604 dicapai dengan Hyperband menggunakan 112 *unit* pada *layer* pertama, *dropout_rate* 0.1, *learning_rate* 0.009132, *optimizer* Adam, *batch_size* 16, dan 64 *unit* pada *layer* kedua.

2) Stacked LSTM

Score 0.562752 dicapai dengan Bayesian Optimization menggunakan 96 *unit* pada *layer* pertama, *dropout_rate* 0.4, *learning_rate* 0.01, *optimizer* Adam, *batch_size* 32, dan 16 *unit* tambahan pada *layer* kedua.

3) LSTM

Score 0.565625, yang dihasilkan oleh metode Optuna menggunakan 16 *unit* pada *layer* pertama, *dropout_rate* 0.5, *learning_rate* 0.008243, *optimizer* Adam, *batch_size* 32, dan tanpa *unit* tambahan pada *layer* kedua.



Gambar 3. Distribusi Score pada Tiap Model

Temuan ini menunjukkan bahwa arsitektur Stacked LSTM cenderung menghasilkan performa lebih baik dibandingkan model lain seperti LSTM atau Bi-LSTM. Hal ini sejalan dengan teori bahwa Stacked LSTM mampu menangkap pola temporal yang lebih kompleks melalui lapisan-lapisan LSTM yang saling bertumpuk. Namun, performa optimal ini sangat bergantung pada pemilihan *hyperparameter* yang tepat, seperti jumlah *unit*, *dropout rate*, dan *learning rate*.

Meskipun model LSTM sederhana memiliki kapasitas representasi yang lebih rendah dibandingkan Stacked LSTM, hasil ini menunjukkan bahwa dengan konfigurasi *hyperparameter* yang tepat, model ini tetap dapat bersaing dalam hal performa. Faktor kunci keberhasilannya adalah penggunaan *dropout rate* yang relatif tinggi (0.5), yang membantu mengurangi *overfitting*, serta *learning rate* yang cukup rendah untuk memastikan konvergensi yang stabil.

Stacked LSTM telah menunjukkan performa terbaik dalam eksperimen, namun terdapat indikasi bahwa pada titik tertentu peningkatan kompleksitas model tidak lagi menghasilkan penurunan *error* secara signifikan. Dalam kondisi tersebut, potensi kombinasi model dapat dijadikan alternatif untuk meningkatkan akurasi secara lebih holistik. Sebagai contoh, menggabungkan prediksi LSTM dengan metode regresi klasik dapat memanfaatkan fitur temporal yang dipelajari oleh LSTM sebagai input tambahan bagi model regresi. Pendekatan hibrida semacam ini memungkinkan integrasi antara pemodelan *deep learning* dan teknik statistik konvensional sehingga dapat mengatasi keterbatasan masing-masing model secara individual.

D. Performa Model Berdasarkan Metode Tuning

Untuk menganalisis performa berdasarkan metode *tuning*, dilakukan dengan metode statistik sebagaimana yang ditampilkan dalam Tabel 2.

Analisis statistik performa tiga metode *tuning hyperparameter* menunjukkan perbedaan signifikan dalam hal konsistensi dan hasil optimal. Optuna mencatatkan rata-rata *score* terendah (0.5865) dan median terendah (0.5813), mengindikasikan bahwa metode ini secara konsisten menghasilkan konfigurasi *hyperparameter* yang lebih optimal. Bayesian Optimization memiliki rata-rata *score* 0.5940 dan median 0.5903, menunjukkan performa baik meskipun sedikit lebih tinggi dibandingkan Optuna. Hyperband menunjukkan rata-rata *score* tertinggi (0.6145) dan median 0.6076, dengan variasi hasil yang lebih besar. Hal ini mungkin disebabkan oleh sifat algoritma Hyperband yang mengurangi alokasi sumber daya untuk konfigurasi yang kurang potensial, sehingga mengorbankan konsistensi.

Analisis konsistensi terhadap metode *tuning*, Optuna memiliki standar deviasi terkecil (0.0140), menandakan hasil yang paling stabil dan konsisten. Bayesian Optimization menunjukkan standar deviasi 0.0198, sedikit lebih variatif dibandingkan Optuna. Sedangkan Hyperband memiliki standar deviasi tertinggi (0.0506), dengan rentang *score* yang sangat lebar (0.5606–1.1822). Hal ini mencerminkan risiko tinggi dalam eksplorasi ruang *hyperparameter*, di mana beberapa konfigurasi menghasilkan performa buruk secara signifikan (misalnya, *score* > 1.0). Visualisasi boxplot pada gambar 4 dapat merepresentasikan hasil analisis konsistensi metode *tuning*.

Metode optimasi *hyperparameter* berbasis *black-box* sebagaimana yang dilakukan oleh penelitian ini, mampu menelusuri ruang pencarian parameter dengan efisiensi tinggi, namun pendekatan ini tidak secara eksplisit memasukkan *constraint* atau batasan riil dari sisi bisnis. Misalnya, biaya penyimpanan bahan baku merupakan faktor kritis yang mempengaruhi profitabilitas, namun metrik yang dioptimalkan seperti MSE dan MAE tidak mampu menggambarkan dampak finansial dari kesalahan prediksi. Dengan demikian, meskipun pendekatan ini efektif dalam meningkatkan akurasi model, terdapat keterbatasan dalam mengintegrasikan pertimbangan ekonomi dan logistik, sehingga diperlukan strategi optimasi multi-obyektif yang juga mengakomodasi *constraint* bisnis untuk menghasilkan solusi yang lebih aplikatif dalam konteks operasional industri.

E. Analisis Performa Model Berdasarkan Kombinasi

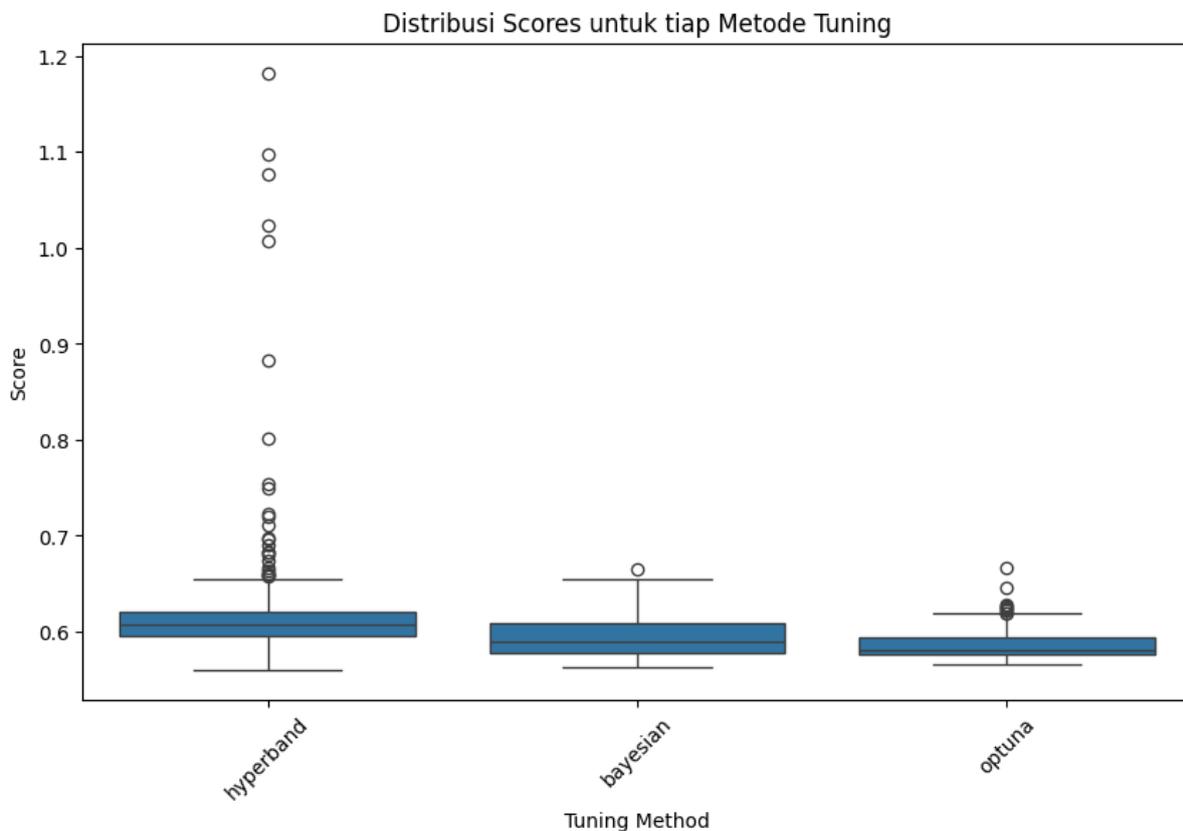
Untuk memahami pola performa model berdasarkan kombinasi metode *tuning* dan tipe model, berikut disajikan tabel 3 serta visualisasi distribusi dari 100 data dengan *score* terendah (performa terbaik) pada gambar 5. Data ini memberikan wawasan mendalam tentang dominasi kombinasi tertentu dalam mencapai hasil optimal.

Tabel 3 dan grafik batang pada Gambar 5 menunjukkan distribusi jumlah kemunculan setiap kombinasi metode *tuning* dan tipe model dari 100 data dengan *score* terendah.

Kombinasi Bayesian Optimization dengan Stacked LSTM mendominasi dengan 49 kemunculan, menunjukkan bahwa metode ini sangat efektif untuk model bertumpuk yang memerlukan eksplorasi *hyperparameter* secara sistematis. Sementara itu, Optuna menunjukkan keunggulan untuk model LSTM sederhana, dengan 37 kemunculan, yang mengindikasikan fleksibilitasnya dalam menemukan konfigurasi optimal untuk model dengan struktur lebih ringkas. Metode Hyperband, meskipun memiliki potensi untuk menemukan skor terbaik, hanya memberikan kontribusi kecil (7 kemunculan) karena pendekatannya yang lebih fokus pada kecepatan dibandingkan presisi.

TABEL II.
ANALISA STATISTIK PERFORMA METODE TUNING

metode tuning	mean	median	std	min	max
bayesian	0.5940	0.5903	0.0198	0.5627	0.6654
hyperband	0.6145	0.6076	0.0506	0.5606	1.1822
optuna	0.5865	0.5813	0.0140	0.5656	0.6660



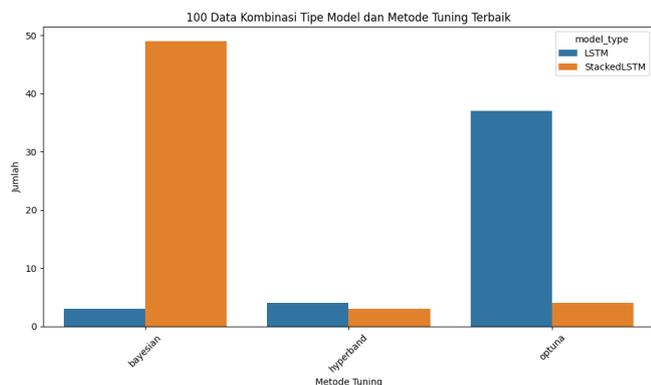
Gambar 4. Distribusi Score pada Metode Tuning

TABEL III.
 DISTRIBUSI 100 SCORE TERENDAH TERHADAP KOMBINASI TIPE
 MODEL DAN METODE TUNING

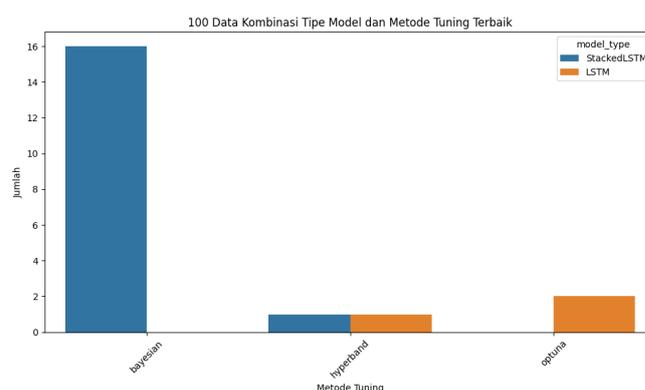
Type Model	Metode Tuning	Jumlah
Stacked LSTM	Bayesian	49
LSTM	Hyperband	4
Stacked LSTM	Hyperband	3
LSTM	Optuna	37
Stacked LSTM	Optuna	4

TABEL IV.
 DISTRIBUSI 20 SCORE TERENDAH TERHADAP KOMBINASI TIPE
 MODEL DAN METODE TUNING

Type Model	Metode Tuning	Jumlah
Stacked LSTM	Bayesian	16
LSTM	Hyperband	1
Stacked LSTM	Hyperband	1
LSTM	Optuna	2



Gambar 5. Distribusi 100 Score Terendah Terhadap Kombinasi Tipe Model dan Metode Tuning



Gambar 6. Distribusi 20 Score Terendah Terhadap Kombinasi Tipe Model dan Metode Tuning

Tabel 4 dan Gambar 6 menampilkan tren yang lebih spesifik terhadap distribusi dari subset 20 data terbaik. Pada subset 20 data terbaik, dominasi Bayesian Optimization dengan Stacked LSTM semakin jelas, dengan 16 kemunculan dari 20 data. Hal ini menunjukkan bahwa kombinasi ini tidak hanya unggul dalam jumlah besar tetapi juga konsisten dalam menghasilkan performa terbaik. Kombinasi lainnya, seperti Optuna dengan LSTM (2 kemunculan) dan Hyperband (2 kemunculan), cenderung tersisih dalam subset ini, mengonfirmasi bahwa mereka kurang kompetitif ketika dipersempit ke performa puncak.

Secara mencolok, model Bi-LSTM tidak muncul sama sekali dalam 20 data terbaik, bahkan dalam subset 100 data terbaik. Model Bi-LSTM menunjukkan performa buruk dapat dikarenakan kompleksitas struktural dan ketidakstabilan *tuning*. Arsitektur dua arahnya membuatnya sensitif terhadap pengaturan *hyperparameter* (seperti *learning rate* dan *dropout*), sehingga rentan *overfitting/underfitting* jika tidak dioptimalkan dengan tepat. Selain itu, Bi-LSTM memiliki variasi skor yang tinggi, menandakan ketidakkonsistenan, terutama saat dikombinasikan dengan metode *tuning* tertentu. Sebaliknya, Stacked LSTM lebih stabil dan mudah dioptimalkan karena lapisan hierarkisnya, memungkinkannya mendominasi data terbaik sementara Bi-LSTM tersisih. Dominasi ini menunjukkan bahwa stabilitas arsitektur dan metode *tuning* yang efektif lebih krusial daripada kompleksitas model dalam mencapai performa optimal.

Arsitektur Bi-LSTM mengolah data dalam dua arah, sehingga diharapkan dapat menangkap konteks temporal yang lebih kaya. Namun, pada data karet alam, yang cenderung menunjukkan pola musiman dan tren *unidirectional*, proses dua arah ini dapat memperkenalkan *noise* dan kompleksitas tambahan yang tidak selalu memberikan manfaat. Ketidakcocokan antara struktur *bidirectional* dengan karakteristik data yang dominan *unidirectional* berpotensi membuat parameter model Bi-LSTM sulit dioptimalkan. Hal ini mengakibatkan sensitivitas *tuning* yang tinggi dan risiko *overfitting* yang lebih besar, sehingga performa prediksi menjadi kurang konsisten.

F. Temuan Kritis dan Diskusi

Penelitian ini menghasilkan berbagai insight yang menarik dari analisis mendalam terhadap dataset tuning hyperparameter yang mencakup eksperimen pada model LSTM, Bi-LSTM, dan Stacked LSTM. Dari analisis deskriptif, terlihat bahwa distribusi nilai *hyperparameter* seperti *epochs*, *units*, *dropout_rate*, *learning_rate*, *batch_size*, dan *units2* memberikan gambaran mengenai sebaran parameter yang digunakan dalam eksperimen, dimana *learning_rate* dan *dropout_rate* menunjukkan peran penting dalam memengaruhi performa model, sebagaimana ditunjukkan oleh korelasi yang signifikan dengan *score*. Data menunjukkan bahwa penurunan *learning_rate* berkorelasi dengan penurunan nilai *score* secara konsisten, sementara peningkatan *dropout_rate* membantu meningkatkan generalisasi model dan mengurangi risiko *overfitting*. Sementara itu, penambahan *units2* di lapisan kedua cenderung meningkatkan *error*, menunjukkan bahwa peningkatan kompleksitas tidak selalu menghasilkan peningkatan performa.

Analisis performa berdasarkan metode *tuning* mengungkap perbedaan menarik antar pendekatan. Metode Optuna menunjukkan konsistensi yang tinggi dengan nilai *score* rata-rata terendah dan variabilitas yang minimal, sedangkan Bayesian dan Hyperband memberikan performa yang lebih variatif. Ketika difokuskan pada subset data terbaik, temuan menunjukkan dominasi kombinasi optimal dari metode Bayesian dengan model Stacked LSTM, yang menghasilkan skor terendah secara konsisten, sedangkan Optuna lebih unggul untuk model LSTM. Dalam subset 100 dan 20 data terbaik, ketidakhadiran model Bi-LSTM juga menjadi sorotan, yang mengindikasikan bahwa dalam konteks eksperimen ini, arsitektur Bi-LSTM mungkin memiliki sensitivitas *tuning* yang lebih tinggi atau kompleksitas yang tidak memungkinkan pencapaian kombinasi *hyperparameter* optimal.

Secara keseluruhan, penelitian ini menyoroti bahwa pemilihan metode *tuning* dan konfigurasi *hyperparameter* memiliki dampak krusial terhadap performa model. Faktor seperti *learning_rate* dan *dropout_rate* muncul sebagai variabel kunci, sedangkan parameter seperti *batch_size* dan *epochs* menunjukkan pengaruh yang lebih terbatas terhadap *error* prediksi. Temuan ini membuka ruang untuk diskusi lebih lanjut mengenai strategi optimasi *hyperparameter*, termasuk pendekatan untuk menyeimbangkan kompleksitas model dengan kebutuhan regularisasi, serta bagaimana perbedaan metode *tuning* dapat diadaptasi berdasarkan jenis arsitektur model yang digunakan.

Temuan penelitian ini memiliki potensi signifikan untuk diimplementasikan pada sistem peramalan di lingkungan produksi nyata. Namun, penerapan model prediksi tidak hanya bergantung pada performa akurasi, melainkan juga membutuhkan integrasi yang mulus dengan sistem informasi yang ada, pemeliharaan kontinuitas data real-time, serta mekanisme *retraining* untuk menyesuaikan perubahan kondisi pasar. Tantangan operasional yang mungkin muncul mencakup masalah latensi dalam pemrosesan data, kesulitan dalam mengukur dan menyesuaikan *constraint* bisnis secara dinamis, serta kebutuhan untuk mengembangkan antarmuka pengguna yang mudah dipahami oleh tim operasional. Diskusi lebih mendalam mengenai bagaimana mengatasi tantangan ini, seperti pengembangan sistem *monitoring* dan *feedback loop*, akan sangat penting untuk menjembatani kesenjangan antara hasil penelitian dan aplikasinya di dunia industri.

IV. KESIMPULAN

Penelitian ini mengevaluasi pengaruh metode *tuning hyperparameter* (Bayesian Optimization, Hyperband, dan Optuna) terhadap performa model LSTM, Bi-LSTM, dan Stacked LSTM dalam meminimalkan nilai *score*. Hasil menunjukkan bahwa Stacked LSTM secara konsisten menjadi model terbaik, dengan menggunakan kombinasi Bayesian Optimization. Konfigurasi optimal melibatkan learning rate sangat rendah (< 0.01), *dropout rate* moderat (0.1–0.4), *optimizer* adaptif (Adam/RMSProp), dan *batch size* kecil (16–32). Sementara itu, Bi-LSTM kurang kompetitif karena sensitivitasnya terhadap *hyperparameter* dan kompleksitas arsitektur dua arah, yang menyebabkan ketidakstabilan performa. Meski Hyperband dan Optuna mampu menemukan skor terbaik dalam kasus tertentu, Bayesian Optimization lebih unggul dalam presisi untuk model kompleks. Temuan ini menegaskan bahwa pemilihan metode *tuning* dan konfigurasi *hyperparameter* harus disesuaikan dengan karakteristik model. Untuk penelitian lanjutan, disarankan eksplorasi metode *hybrid* (gabungan Bayesian Optimization dan Hyperband) serta optimasi Bi-LSTM dengan pendekatan lebih hati-hati untuk meningkatkan stabilitasnya.

DAFTAR PUSTAKA

- [1] N. P. Setyadi, "Penerapan Data Mining Untuk Prediksi Hasil Produksi Karet Menggunakan Algoritma Decision Tree C4.5," *Jurnal Teknologipintar.org*, vol. 2, no. 7, 2022, Accessed: Dec. 15, 2023. [Online]. Available: <http://teknologipintar.org/index.php/teknologipintar/article/view/203/201>
- [2] A. Mustakim, Suharno, and Burhanuddin, "Hubungan Penerapan Best Management Practice Pengolahan Karet Remah Pada Kinerja Pemasaran PT. Remco Rubber Indonesia," *Forum Agribisnis*, vol. 13, no. 2, 2023, doi: 10.29244/fagb.13.2.240-247.
- [3] L. F. Syarif, D. S. Agustina, A. Alamsyah, I. S. Nugraha, and H. Asywardi, "OUTLOOK KOMODITAS KARET ALAM INDONESIA 2023," *Jurnal Penelitian Karet*, 2023, doi: 10.22302/ppk.jpk.v4i1i.841.
- [4] H. Herudin, E. Yurisinthae, and A. Suyatno, "Konversi Usahatani Karet Menjadi Usahatani Kelapa Sawit Kecamatan Belitang Hilir Kabupaten Sekadau," *Jurnal Sosial Ekonomi Pertanian*, vol. 18, no. 1, 2021, doi: 10.20956/jssep.v18i1.18459.
- [5] E. Amrina and A. Y. Dewi, "Optimizing Inventory Control System of Crumb Rubber Raw Material: a Case Study," *IOP Conf Ser Mater Sci Eng*, vol. 1041, no. 1, p. 012045, Jan. 2021, doi: 10.1088/1757-899x/1041/1/012045.
- [6] I. H. L. Group, "True Cost of Out of Stocks and Over Stocks," 2023. [Online]. Available: https://4153407.fs1.hubspotusercontent-na1.net/hubfs/4153407/IHL%20Group%20-%20True%20Cost%20of%20Out%20of%20Stocks%20and%20Over%20Stocks.pdf?__hstc=&__hssc=&hscCtaTracking=8aceb1df-5e38-48df-86c6-8e7e871935f7%7C61e2db49-fec5-42d1-bed4-1f43f981aca3
- [7] J. Amar, S. Rahimi, Z. Surak, and N. von Bismarck, "AI-driven operations forecasting in data-light environments," <https://www.mckinsey.com/capabilities/operations/our-insights/ai-driven-operations-forecasting-in-data-light-environments>.
- [8] Data Bridge Market Research, "Global Natural Rubber Market Size, Share, and Trends Analysis Report – Industry Overview and Forecast to 2032." Accessed: Mar. 07, 2025. [Online]. Available: <https://www.databridgemarketresearch.com/reports/global-natural-rubber-market?srsltid=AfmBOoolxEZYZGpM12mccUpokIMIU4Drbe1rxB48ON4FD2RHZbxKM8E>
- [9] P. A. Qori, D. S. Oktafani, and I. Kharisudin, "Analisis Peramalan dengan Long Short Term Memory pada Data Kasus Covid-19 di Provinsi Jawa Tengah," *PRISMA, Prosiding Seminar Nasional Matematika*, vol. 5, 2022.
- [10] K. Kwanda, D. E. Herwindiati, and M. D. Lauro, "Perbandingan LSTM dan Bidirectional LSTM pada Sistem Prediksi Harga Saham Berbasis Website," *Ranah Research : Journal of Multidisciplinary Research and Development*, vol. 7, no. 1, pp. 26–35, Nov. 2024.
- [11] K. Qiu, J. Li, and D. Chen, "Optimized long short-term memory (LSTM) network for performance prediction in unconventional reservoirs," *Energy Reports*, vol. 8, pp. 15436–15445, Nov. 2022, doi: 10.1016/j.egy.2022.11.130.

- [12] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," *Informatics*, vol. 8, no. 4, 2021, doi: 10.3390/informatics8040079.
- [13] S. Hanifi, A. Cammarono, and H. Zare-Behtash, "Advanced hyperparameter optimization of deep learning models for wind power prediction," *Renew Energy*, vol. 221, 2024, doi: 10.1016/j.renene.2023.119700.
- [14] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," 2023. doi: 10.1002/widm.1484.
- [15] J. P. Lai, Y. L. Lin, H. C. Lin, C. Y. Shih, Y. P. Wang, and P. F. Pai, "Tree-Based Machine Learning Models with Optuna in Predicting Impedance Values for Circuit Analysis," *Micromachines (Basel)*, vol. 14, no. 2, 2023, doi: 10.3390/mi14020265.
- [16] P. Foroutan and S. Lahmiri, "Deep learning systems for forecasting the prices of crude oil and precious metals," *Financial Innovation*, vol. 10, no. 1, p. 111, 2024, doi: 10.1186/s40854-024-00637-z.
- [17] Q. Wen *et al.*, "Transformers in Time Series: A Survey," in *IJCAI International Joint Conference on Artificial Intelligence*, 2023. doi: 10.24963/ijcai.2023/759.
- [18] A. M. Vincent and P. Jidesh, "An improved hyperparameter optimization framework for AutoML systems using evolutionary algorithms," *Sci Rep*, vol. 13, no. 1, 2023, doi: 10.1038/s41598-023-32027-3.
- [19] A. Karras, C. Karras, N. Schizas, M. Avlonitis, and S. Sioutas, "AutoML with Bayesian Optimizations for Big Data Management," *Information (Switzerland)*, vol. 14, no. 4, 2023, doi: 10.3390/info14040223.
- [20] Y. Zhao, W. Zhang, and X. Liu, "Grid search with a weighted error function: Hyper-parameter optimization for financial time series forecasting," *Appl Soft Comput*, vol. 154, 2024, doi: 10.1016/j.asoc.2024.111362.
- [21] A. J. Fetterman *et al.*, "Tune As You Scale: Hyperparameter Optimization For Compute Efficient Training," 2023. [Online]. Available: <https://arxiv.org/abs/2306.08055>
- [22] M. Aach, R. Sarma, H. Neukirchen, M. Riedel, and A. Lintermann, "Resource-Adaptive Successive Doubling for Hyperparameter Optimization with Large Datasets on High-Performance Computing Systems," 2024. [Online]. Available: <https://arxiv.org/abs/2412.02729>
- [23] H. Dhake, Y. Kashyap, and P. Kosmopoulos, "Algorithms for Hyperparameter Tuning of LSTMs for Time Series Forecasting," *Remote Sens (Basel)*, vol. 15, no. 8, 2023, doi: 10.3390/rs15082076.
- [24] R. A. Andhika Viadinugroho and D. Rosadi, "A weighted metric scalarization approach for multiobjective BOHB hyperparameter optimization in LSTM model for sentiment analysis," *Inf Sci (N Y)*, vol. 644, 2023, doi: 10.1016/j.ins.2023.119282.
- [25] X. Xiong and G. Qing, "A hybrid day-ahead electricity price forecasting framework based on time series," *Energy*, vol. 264, 2023, doi: 10.1016/j.energy.2022.126099.
- [26] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," 2022. doi: 10.5194/gmd-15-5481-2022.
- [27] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput Sci*, vol. 7, 2021, doi: 10.7717/PEERJ-CS.623.