

# Peramalan Penerimaan Iuran Program Jaminan Kehilangan Pekerjaan BPJS Ketenagakerjaan Menggunakan Metode *Hybrid Prophet-BiLSTM*

**Muhammad Nashih Ulwan<sup>1</sup>, Danardono<sup>2</sup>, Ajrul Azwar<sup>3</sup>**

<sup>1,2</sup>Universitas Gadjah Mada, <sup>3</sup>BPJS Ketenagakerjaan

Email: [muhammad.nashih.ulwan@mail.ugm.ac.id](mailto:muhammad.nashih.ulwan@mail.ugm.ac.id)

## Abstrak

BPJS Ketenagakerjaan menyelenggarakan program Jaminan Kehilangan Pekerjaan (JKP) sebagai bentuk perlindungan sosial bagi pekerja yang mengalami pemutusan hubungan kerja. Fluktuasi penerimaan iuran JKP membutuhkan metode peramalan yang akurat sebagai dasar perencanaan keuangan untuk keberlanjutan program. Penelitian ini bertujuan untuk mengembangkan model peramalan penerimaan iuran JKP menggunakan pendekatan *Hybrid Prophet- Bidirectional Long Short-Term Memory* (BiLSTM) yang dikombinasikan melalui *Linear Optimal Weighting Estimator* (LOWE). Data yang digunakan berupa data bulanan iuran JKP periode Februari 2021 hingga Agustus 2025. Melalui evaluasi kinerja model menggunakan *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan *Mean Absolute Percentage Error* (MAPE), dihasilkan bahwa model *Hybrid Prophet*-BiLSTM memberikan performa prediksi yang lebih baik dibandingkan model *Prophet* dan BiLSTM secara individual, khususnya pada data uji. Peramalan iuran JKP pada periode September 2025 hingga Agustus 2026 menunjukkan pola musiman yang konsisten dengan tren historis. Temuan ini menunjukkan bahwa pendekatan *Hybrid Prophet*-BiLSTM dengan pembobotan LOWE dalam konteks peramalan iuran JKP, yang memberikan alternatif metode peramalan yang lebih adaptif dan akurat bagi pengambilan keputusan strategis BPJS Ketenagakerjaan.

## Kata kunci:

BPJS Ketenagakerjaan, Jaminan Kehilangan Pekerjaan, Model *Hybrid Prophet*-BiLSTM,  
Perlindungan Sosial

## Pendahuluan

BPJS Ketenagakerjaan merupakan lembaga yang menyelenggarakan program jaminan sosial tenaga kerja di Indonesia, salah satunya adalah program Jaminan Kehilangan Pekerjaan (JKP) yang mana merupakan program terbaru dan memiliki keterbatasan data historis dibandingkan program lainnya. Program ini memberikan perlindungan bagi pekerja yang mengalami pemutusan hubungan kerja (PHK) melalui pemberian manfaat uang tunai, akses informasi pasar kerja, serta pelatihan keterampilan kerja. Berdasarkan Peraturan Pemerintah

---

Nomor 6 Tahun 2025 tentang Perubahan atas Peraturan Pemerintah Nomor 37 Tahun 2021 tentang Penyelenggaraan Program Jaminan Kehilangan Pekerjaan, bahwa program ini dibiayai melalui rekomposisi iuran dari program Jaminan Kecelakaan Kerja (JKK), serta dukungan anggaran dari pemerintah pusat (Republik Indonesia 2025). Mengingat pentingnya keberlangsungan dana untuk program JKP, diperlukan perencanaan keuangan yang akurat, terutama dalam memproyeksikan jumlah iuran yang akan diterima di masa depan. Hal ini menjadi semakin krusial dalam konteks dinamika ekonomi dan perubahan kebijakan ketenagakerjaan yang dapat memengaruhi jumlah iuran secara signifikan.

Berdasarkan data dari BPJS Ketenagakerjaan, secara umum, pola pergerakan iuran menunjukkan kecenderungan meningkat dari tahun ke tahun, khususnya sepanjang 2021 hingga 2024. Kenaikan ini mencerminkan adanya perluasan cakupan kepesertaan serta peningkatan kepatuhan pemberi kerja dengan rata-rata pertumbuhan tahunan (*Year-on-Year*) yang stabil di kisaran 10%. Selain tren jangka panjang yang positif, terlihat pula adanya fluktuasi musiman yang konsisten setiap tahunnya, yang ditandai dengan lonjakan penerimaan pada bulan Desember (mencapai 20%) dan penurunan tajam secara administratif pada bulan Januari (mencapai -32%).

Namun demikian, memasuki awal tahun 2025, nilai iuran cenderung menurun dan tidak menunjukkan pola peningkatan seperti tahun-tahun sebelumnya, di mana angka pertumbuhan YoY terkontraksi hingga menyentuh -7,57% pada Agustus 2025. Fenomena ini disebabkan oleh adanya perubahan regulasi terkait penghentian atau penyesuaian rate rekomposisi iuran dari program Jaminan Kematian (JKM) yang sebelumnya dialokasikan untuk mendanai JKP. Mengingat JKP merupakan program yang relatif baru dengan ketersediaan data historis yang terbatas, adanya *structural break* akibat perubahan kebijakan ini menciptakan tantangan besar dalam memproyeksikan keberlanjutan dana jaminan sosial di masa depan.

Peramalan berbasis data historis menjadi pendekatan yang banyak digunakan dalam memprediksi nilai iuran program jaminan sosial. Karakteristik data iuran JKP yang bersifat fluktuatif, musiman, serta dipengaruhi oleh faktor nonlinier menjadikan metode statistik klasik seperti ARIMA sering kali tidak mampu menangkap kompleksitas pola datanya secara optimal. Oleh karena itu, berbagai pendekatan modern telah dikembangkan, seperti model *Prophet* yang dikembangkan oleh Facebook, yang unggul dalam menangkap tren dan musiman serta relatif mudah digunakan (Shakeel et al. 2023; Kwarteng 2024). *Prophet* terbukti memberikan performa prediksi yang stabil pada data dengan struktur musiman dan tren yang berubah-ubah. Efektivitas *Prophet* dalam menangani data musiman telah divalidasi dalam berbagai domain,

mulai dari peramalan polusi udara, kasus pandemi, hingga fluktuasi harga aset kripto (Samal et al. 2019; Papastefanopoulos et al. 2020; Yenidoğan et al. 2018).

Sementara itu, pendekatan berbasis *deep learning* seperti *Long Short-Term Memory* (LSTM) dan pengembangannya, *Bidirectional LSTM* (BiLSTM), telah banyak diterapkan untuk memodelkan data deret waktu dengan pola nonlinier yang kompleks (Menculini et al. 2022). Keunggulan BiLSTM terletak pada kemampuannya menangkap pola data dari dua arah (masa lalu dan masa depan), sehingga lebih fleksibel dalam memahami dinamika data ketenagakerjaan yang bersifat sekuensial dan tidak selalu linier. Penggabungan *Prophet* dan LSTM atau BiLSTM dalam bentuk model *hybrid* telah banyak dieksplorasi pada berbagai konteks, seperti peramalan konsumsi energi, permintaan listrik, dan debit air sungai (Arslan et al. 2022; Kenyi et al. 2024; Choubey et al. 2023).

Meskipun pendekatan *Hybrid Prophet-LSTM* telah digunakan dalam berbagai studi, hingga saat ini belum ditemukan studi yang secara spesifik menerapkan kombinasi *Prophet* dan BiLSTM untuk peramalan jumlah iuran JKP di Indonesia, khususnya dengan pendekatan pembobotan menggunakan metode *Linear Optimal Weighting Estimator* (LOWE). Metode ini memungkinkan model menggabungkan hasil prediksi *Prophet* dan BiLSTM secara linier berdasarkan performa relatif masing-masing model, sehingga dapat meningkatkan akurasi dan stabilitas hasil peramalan. Dengan demikian, artikel ini menawarkan kebaruan ilmiah dalam bentuk pengembangan dan implementasi model *Hybrid Prophet-BiLSTM* yang dikombinasikan melalui metode pembobotan LOWE dalam konteks peramalan iuran program JKP.

Permasalahan utama dalam kajian ini adalah bagaimana mengembangkan model prediktif yang mampu menangkap dinamika musiman, tren jangka panjang, serta komponen nonlinier dalam data iuran JKP, sekaligus menghasilkan prediksi yang lebih akurat dibandingkan metode tunggal. Oleh karena itu, hipotesis dari penelitian ini adalah bahwa model *Hybrid Prophet-BiLSTM* yang dikombinasikan melalui pembobotan LOWE akan memberikan performa prediksi yang lebih baik dibandingkan dengan model *Prophet* atau BiLSTM secara tunggal.

Tujuan dari artikel ini adalah untuk mengembangkan dan mengevaluasi model *Hybrid Prophet-BiLSTM* dalam peramalan jumlah iuran JKP. Pada penelitian ini data yang digunakan adalah data penerimaan iuran JKP tanpa variabel lainnya. Evaluasi dilakukan dengan membandingkan performa prediksi model *hybrid* terhadap model tunggal menggunakan metrik evaluasi yaitu MSE, RMSE, MAE, dan MAPE. Hasil dari kajian ini diharapkan dapat memberikan kontribusi terhadap pengembangan metode prediksi deret waktu dalam konteks jaminan sosial

ketenagakerjaan, serta menjadi masukan strategis bagi BPJS Ketenagakerjaan dalam menyusun kebijakan pengelolaan dana jangka menengah hingga panjang.

## Metode Penelitian

Dalam penelitian ini, pendekatan yang digunakan untuk pemodelan dan prediksi data runtun waktu adalah kombinasi metode *Prophet* dan *Bidirectional Long Short-Term Memory* (BiLSTM), yang kemudian digabungkan secara optimal dengan metode *Linear Optimal Weighting Estimator* (LOWE). Pemilihan metode ini didasarkan pada kompleksitas data runtun waktu yang memuat komponen linier dan nonlinier.

### 1. Model *Prophet*

Model *Prophet* merupakan model peramalan runtun waktu yang dikembangkan oleh tim *data science* Facebook dan didasarkan pada pendekatan dekomposisi aditif yang mencakup tiga komponen utama, yaitu tren, musiman (menggambarkan perilaku data nonlinear secara harian, bulanan, atau tahunan), dan dampak hari libur yang meningkatkan akurasi model (Almazrouee et al. 2020; Arslan 2022). Secara umum, model *Prophet* dinyatakan sebagai::

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (1.1)$$

di mana  $y(t)$  adalah nilai observasi (nilai aktual),  $g(t)$  merepresentasikan tren,  $s(t)$  merepresentasikan musiman, dan  $h(t)$  merepresentasikan pengaruh hari libur, dan  $\varepsilon_t$  adalah *error*.

Komponen tren  $g(t)$  dimodelkan menggunakan fungsi waktu linier maupun nonlinier, termasuk model pertumbuhan logistik dan *piecewise linear* dengan titik perubahan (*changepoints*). Bentuk dasar pertumbuhan logistik dinyatakan sebagai:

$$g(t) = \frac{C}{1 + \exp(k(t - m))} \quad (1.2)$$

dengan  $C$  mewakili kapasitas maksimum,  $k$  mewakili laju pertumbuhan, dan  $m$  menunjukkan parameter keseimbangan. Ketika variabel laju  $k$  diubah, variabel keseimbangan juga harus disesuaikan agar urutan tetap terhubung. Model pertumbuhan logistik berbasis segmen ditunjukkan sebagai berikut:

$$g(t) = \frac{C(t)}{1 + \exp(k + a(t)^T \delta)(t - m + a(t)^T \gamma)} \quad (1.3)$$

di mana  $\delta$  dan  $\gamma$  adalah vektor perubahan laju pertumbuhan, yaitu perubahan laju yang terjadi pada waktu tertentu. Titik perubahan dihasilkan oleh fenomena yang menyebabkan perubahan laju pertumbuhan, dan tren model dinyatakan sebagai:

$$g(t) = (k + a(t)^T \delta)(t - m + a(t)^T \gamma) \quad (1.4)$$

di mana  $k$  mewakili laju pertumbuhan,  $m$  adalah parameter *offset*,  $\delta$  adalah perubahan laju, dan  $\delta$  adalah titik perubahan tren yang ditetapkan ke  $s_j\delta_j$  untuk membuat fungsi tetap kontinu.

Komponen musiman  $s(t)$  dimodelkan menggunakan deret Fourier untuk menangkap pola periodik yang bersifat nonlinier, dan dinyatakan sebagai:

$$s(t) = \sum_{n=1}^N \left( a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right) \quad (1.5)$$

di mana  $P$  adalah periode musiman, dengan  $P = 365.25$  untuk musiman tahunan,  $P = 30$  untuk musiman bulanan, dan  $P = 7$  untuk musiman mingguan.

Komponen hari libur  $h(t)$  digunakan untuk memodelkan pengaruh hari libur dan kejadian khusus yang tidak mengikuti pola periodik, dengan memasukkan matriks tanggal dan karakteristik hari libur yang ditentukan oleh pengguna.

Model *Prophet* dirancang sebagai model dengan penyetelan parameter otomatis, namun beberapa *hyperparameter* utama, seperti jumlah titik perubahan dan komponen musiman, dapat disesuaikan untuk meningkatkan akurasi prediksi. Dalam penelitian ini, model *Prophet* dibangun menggunakan parameter optimal, termasuk musiman harian dan mingguan, serta pemilihan titik perubahan yang sesuai dengan karakteristik data.

## 2. Bidirectional LSTM (BiLSTM)

BiLSTM merupakan pengembangan dari LSTM konvensional yang dirancang untuk menangkap informasi sekuensial dari dua arah, yaitu maju (*forward*) dan mundur (*backward*), sehingga mampu memanfaatkan informasi masa lalu dan masa depan secara simultan dalam satu jendela input. Kemampuan ini memungkinkan BiLSTM untuk menangkap pola dan dependensi temporal yang lebih kompleks, termasuk korelasi yang tidak hanya berasal dari urutan masa lalu, tetapi juga dari struktur keseluruhan window historis (Schuster & Paliwal, 1997). Pendekatan ini memungkinkan BiLSTM menangkap pola dan dependensi temporal yang lebih kompleks dibandingkan LSTM satu arah, khususnya ketika hubungan antar titik waktu bersifat nonlinier dan melibatkan keterkaitan jangka panjang (Althelaya, El-Alfy, dan Mohammed 2018). Modifikasi arsitektur BiLSTM dengan metode integrasi lainnya juga telah terbukti meningkatkan akurasi diagnosa dan prediksi beban energi yang sangat volatil (Li et al. 2022; Yan et al. 2018).

Pada BiLSTM, satu lapisan LSTM memproses data mengikuti urutan waktu, sedangkan lapisan lainnya memproses data secara terbalik. Output dari masing-masing lapisan berupa skor prediksi, yang kemudian dihitung untuk menjadi output akhir dengan bobot tertentu (Shiri et al. 2023). Karena struktur dari BiLSTM masih mengacu pada LSTM, maka pembelajaran dilakukan melalui beberapa *gate*. Setiap *gate* memiliki fungsi berbeda untuk mengatasi masalah *vanishing* dan *exploding gradient*.

$$\text{Forward LSTM: } h_t \rightarrow = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1.6)$$

$$\text{Backward LSTM: } h_t \leftarrow = H(W_{xh}x_t + W_{hh}h_{t+1} + b_h) \quad (1.7)$$

$$\text{Combined Output: } y_t \Rightarrow = H(W_{h \rightarrow y}h_{\rightarrow} + W_{h \leftarrow y}h_{\leftarrow} + b_y) \quad (1.8)$$

*Conveyor belt* dalam LSTM terdiri dari dua komponen utama dan tiga *gate*. Bagian atas dari LSTM disebut sebagai *long-term memory* atau *cell state*, sedangkan bagian bawahnya disebut sebagai *short-term memory* atau *hidden state*. Aliran dimulai dari *forget gate*, diikuti dengan *input gate* yang terhubung ke *cell memory* baru, dan diakhiri dengan *output gate* (Ghojogh dan Ghodsi 2023).

Langkah pertama dalam proses LSTM adalah *forget gate*. *Forget gate* mengevaluasi apakah suatu nilai relevan atau tidak dengan memberikan bobot yang sesuai. Mekanisme ini mengatur seberapa banyak informasi dari memori sebelumnya yang akan diteruskan.

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + (p_f \odot c_{t-1}) + b_f) \quad (1.9)$$

dengan  $W_f \in R^{p \times p}$ ,  $U_f \in R^{p \times d}$ , dan  $b_f \in R^p$ . *Peephole weight* dinyatakan sebagai  $p_f \in R^p$ . Langkah selanjutnya adalah proses pembentukan *new memory cell*. *New memory cell* adalah *gate* dengan aktivasi *tanh* yang menghasilkan memori baru yang disebut  $\tilde{c}_t$ :

$$\tilde{c}_t = \tanh(W_c h_{t-1} + U_c x_t + b_c) \quad (1.10)$$

dengan  $W_c \in R^{p \times p}$ ,  $U_c \in R^{p \times d}$ , dan  $b_c \in R^p$ . Kemudian terdapat *Input gate* yang berfungsi sebagai penyaring bagian mana dari *new memory vector* yang akan disimpan:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + (p_i \odot c_{t-1}) + b_i) \quad (1.11)$$

dengan  $W_i \in R^{p \times p}$ ,  $U_i \in R^{p \times d}$ ,  $b_i \in R^p$ , dan  $p_i \in R^p$ . Terakhir, terdapat *output gate* yang berfungsi untuk mengambil informasi dari memori jangka panjang dan mentransformasikannya melalui fungsi *tanh* untuk menghasilkan:

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + (p_o \odot c_t) + b_o) \quad (1.12)$$

dengan  $W_o \in R^{p \times p}$ ,  $U_o \in R^{p \times d}$ ,  $b_o \in R^p$ , dan  $p_o \in R^p$ .

Dengan memanfaatkan dua arah pemrosesan dan mekanisme *gate* tersebut, BiLSTM mampu menghasilkan representasi sekuensial yang lebih kaya dan stabil, sehingga efektif digunakan untuk meningkatkan akurasi prediksi pada data runtun waktu.

### 3. **Hyperparameter Tuning dengan Grid Search**

Salah satu algoritma yang dapat digunakan untuk menentukan parameter optimal pada model-model machine learning dan *deep learning* adalah algoritma *grid search*. Pada algoritma ini akan dilakukan pembagian jangkauan parameter yang akan dioptimalkan ke dalam grid dan melintasi semua titik untuk mendapatkan parameter optimal (Prahutama et al., 2014). Metode *grid search* dilakukan dengan mencoba kombinasi parameter satu per satu dan membandingkan nilai terbaik yang diberikan oleh parameter tersebut. Dalam aplikasinya, algoritma *grid search* harus dipadukan dengan beberapa metrik evaluasi untuk mengetahui performa dari setiap kombinasinya.

*Hyperparameter tuning* adalah proses menemukan serangkaian *hyperparameter* optimal yang menghasilkan kinerja terbaik untuk model *machine learning*. Proses ini sangat penting karena *hyperparameter* mengontrol proses pembelajaran dan struktur model, seperti tingkat pembelajaran, jumlah neuron dalam jaringan saraf, atau ukuran *kernel* dalam *support vector machine*, yang secara langsung memengaruhi kinerjanya (Yang dan Shami 2020). Berbeda dengan parameter model, yang dipelajari dari data, *hyperparameter* diatur sebelum pelatihan dan memerlukan pemilihan yang cermat (Goodfellow, Bengio, dan Courville 2016). *Hyperparameter tuning* dapat meningkatkan kinerja dan generalisasi model (Claesen dan De Moor 2015).

*Hyperparameter* memainkan peran krusial dalam kinerja berbagai model *machine learning*. Dalam jaringan saraf tiruan, *hyperparameter* seperti tingkat pembelajaran, ukuran *batch*, jumlah lapisan, dan jumlah unit per lapisan secara signifikan memengaruhi kinerja model (Goodfellow, Bengio, dan Courville 2016). Penyetelan yang tepat dari *hyperparameter* ini dapat menghasilkan konvergensi yang lebih cepat dan akurasi yang lebih tinggi (Pavitha dan Sugave 2023). Terlebih lagi, dengan mengidentifikasi *hyperparameter* yang optimal, sumber daya komputasi digunakan lebih efisien, mengurangi waktu dan biaya pelatihan. Efisiensi ini sangat penting untuk model dan kumpulan data skala besar (Bates et al., 2023). Penyetelan parameter ini meningkatkan kemampuan model untuk menangani data yang tidak dapat dipisahkan secara linier.

### 4. **Model Hybrid Prophet-BiLSTM**

Data *time series* memiliki hubungan linier dan nonlinier. Metode statistik efektif untuk menangani hubungan linier, namun tidak mampu mengatasi hubungan nonlinier (Martínez et al. 2018). Sebaliknya, pendekatan *neural network* dapat memodelkan keduanya, meski membutuhkan pemilihan parameter yang tepat dan waktu pelatihan yang

panjang. Keunggulan utama model *neural network (NN)* adalah kemampuannya menangani hubungan nonlinier dengan efektif.

Model *hybrid* digunakan untuk mengatasi keterbatasan model tunggal dengan menggabungkan keunggulan dari dua pendekatan yang berbeda. Dalam studi ini, model *Prophet* digunakan untuk menangkap pola *trend* dan *seasonality* dalam data, sementara model *bidirectional LSTM* (BiLSTM) digunakan untuk mempelajari pola-pola nonlinier dan kompleksitas hubungan temporal dalam data *residual* yang tidak dapat dimodelkan secara eksplisit oleh *Prophet*. Kombinasi kedua model ini diharapkan mampu meningkatkan akurasi prediksi dengan memanfaatkan kekuatan *Prophet* dalam mengidentifikasi pola musiman serta keunggulan BiLSTM dalam menangkap dinamika jangka panjang dan dependensi waktu yang tersembunyi.

Untuk menggabungkan hasil prediksi dari kedua model tersebut secara optimal, digunakan pendekatan *Linear Optimal Weighting Estimator* (LOWE) sebagaimana diperkenalkan oleh Chahkotahi dan Khashei (2022). LOWE merupakan metode non-iteratif yang berbasis regresi linier tanpa intersep, yang bertujuan meminimalkan selisih kuadrat antara nilai aktual dan gabungan prediksi dari dua model. Secara matematis, hasil prediksi gabungan pada waktu ke- $t$  dinyatakan sebagai:

$$\hat{y}_t^{Hybrid} = \alpha \cdot \hat{y}_t^{(1)} + \beta \cdot \hat{y}_t^{(2)} \quad (1.13)$$

dengan  $\hat{y}_t^{(1)}$  dan  $\hat{y}_t^{(2)}$  merupakan hasil prediksi dari dua model berbeda, serta  $\alpha$  dan  $\beta$  merupakan bobot yang diberikan pada masing-masing model. Agar kombinasi lebih terkendali dan interpretatif, diberlakukan kendala:

$$\alpha + \beta = 1, \quad \alpha, \beta \in [0,1] \quad (1.14)$$

Dengan adanya *constraint* tersebut, persamaan prediksi dapat disederhanakan menjadi:

$$\hat{y}_t^{Hybrid} = \alpha \cdot \hat{y}_t^{(1)} + (1 - \alpha) \cdot \hat{y}_t^{(2)} \quad (1.15)$$

Penentuan nilai optimal dari  $\alpha$  dilakukan dengan pendekatan *Linear Optimal Weighting Estimator* (LOWE) yang bersifat non-iteratif. Tujuan dari metode ini adalah untuk meminimalkan jumlah kuadrat *error* antara nilai aktual dan hasil prediksi gabungan, sebagaimana ditunjukkan dalam fungsi objektif berikut:

$$\min_{\alpha} \sum_{t=1}^n (y_t - [\alpha \cdot \hat{y}_t^{(1)} + (1 - \alpha) \cdot \hat{y}_t^{(2)}])^2 \quad (1.16)$$

Metode ini merupakan bentuk khusus dari regresi linier tanpa intersep, yang dapat diselesaikan secara analitik menggunakan pendekatan *least squares*. Misalkan dibentuk matriks desain  $X$  dan vektor target  $Y$  sebagai berikut:

$$X = [\hat{y}_1^{(1)} \hat{y}_1^{(2)} \hat{y}_2^{(1)} \hat{y}_2^{(2)} \dots \hat{y}_n^{(1)} \hat{y}_n^{(2)}], \quad Y = [y_1 \ y_2 \ \dots \ y_n]. \quad (1.17)$$

Bobot optimal  $W = [\alpha, \beta]^T$  diperoleh dengan menyelesaikan persamaan berikut:

$$W = (X^T X)^{-1} X^T Y. \quad (1.18)$$

## 5. Validasi Model

Evaluasi performa model peramalan deret waktu memerlukan skema validasi yang mempertahankan struktur temporal data. Berbeda dengan *cross-validation* konvensional yang mengacak urutan data, validasi deret waktu harus mengikuti urutan kronologis agar mencerminkan kondisi nyata, di mana peramalan masa depan bergantung pada data masa lalu (Bergmeir dan Benítez 2012). Oleh karena itu, penelitian ini menggunakan dua pendekatan validasi berurutan yang umum, yaitu *rolling window* dan *expanding window* (Cerqueira, Torgo, dan Mozetič 2020).

Metode *rolling window* merupakan teknik validasi yang menggunakan *subset* data latih dengan ukuran tetap. Setelah proses pelatihan dan peramalan selesai pada satu iterasi, jendela data digeser maju satu periode, sehingga observasi paling awal dikeluarkan dari jendela dan satu observasi baru ditambahkan. Proses ini diulangi secara berurutan hingga seluruh data yang tersedia digunakan. Dengan mempertahankan ukuran jendela yang konstan, metode ini memungkinkan model lebih adaptif terhadap pola data terbaru yang mungkin berubah dari waktu ke waktu.

Secara matematis, misalkan tersedia data deret waktu  $Y = \{y_1, y_2, \dots, y_T\}$ . Untuk iterasi ke- $i$  dan ukuran jendela sebesar  $w$ , maka data pelatihan dan pengujian pada iterasi tersebut masing-masing didefinisikan sebagai:

$$D_{train,i} = \{y_i, y_{i+1}, \dots, y_{i+w-1}\} \quad (1.19)$$

$$D_{test,i} = \{y_{i+w}\}. \quad (1.20)$$

Berdasarkan metodologi yang telah diuraikan, penelitian ini mengintegrasikan model *Prophet* yang digunakan untuk mengidentifikasi tren dan pola musiman (Taylor dan Letham 2018), bersama BiLSTM untuk memodelkan hubungan nonlinier dan dependensi temporal yang kompleks, khususnya pada komponen residual (Siami-Namini, Tavakoli, dan Namin 2019). Kedua pendekatan tersebut kemudian digabungkan melalui *Linear Optimal Weighting Estimator* (LOWE) guna memperoleh prediksi yang optimal. Proses pemodelan dilengkapi dengan penyetelan *hyperparameter* serta skema validasi berurutan yang mempertahankan struktur temporal data, sehingga evaluasi kinerja model mencerminkan kondisi peramalan yang realistik. Selanjutnya, bagian pembahasan

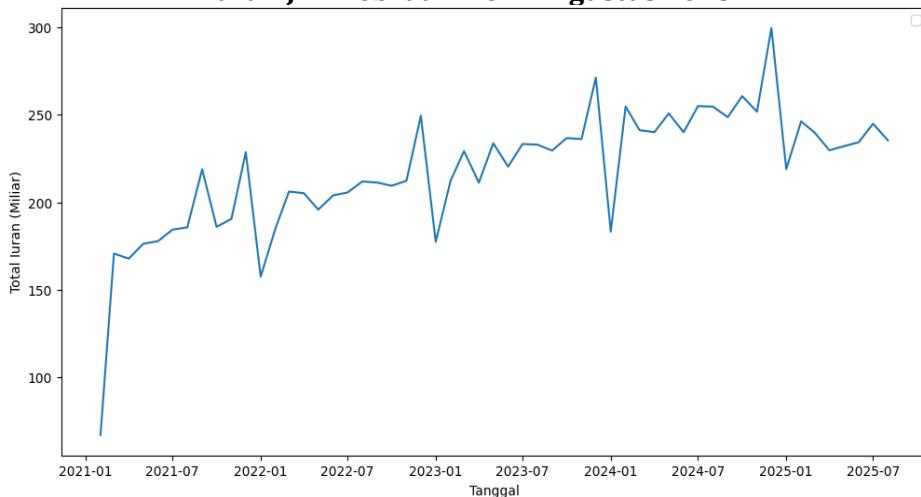
menyajikan hasil empiris dari penerapan model-model tersebut serta implikasinya terhadap tujuan penelitian.

## Pembahasan

Data yang digunakan dalam penelitian ini merupakan data Iuran Jaminan Kehilangan Pekerjaan (JKP) yang dikelola oleh BPJS Ketenagakerjaan. Dataset ini terdiri atas 42 titik waktu dengan frekuensi bulanan, yang mencakup periode mulai dari Februari 2021 hingga Agustus 2025. Periode tersebut dipilih berdasarkan ketersediaan data pada saat proses permintaan data dilakukan secara resmi kepada pihak terkait. Keterbatasan cakupan waktu ini menjadi salah satu tantangan dalam penelitian, namun tetap dapat dianggap representatif untuk membangun model prediksi karena telah mencakup lebih dari tiga tahun data historis yang mencerminkan pola-pola yang terdapat pada data iuran JKP.

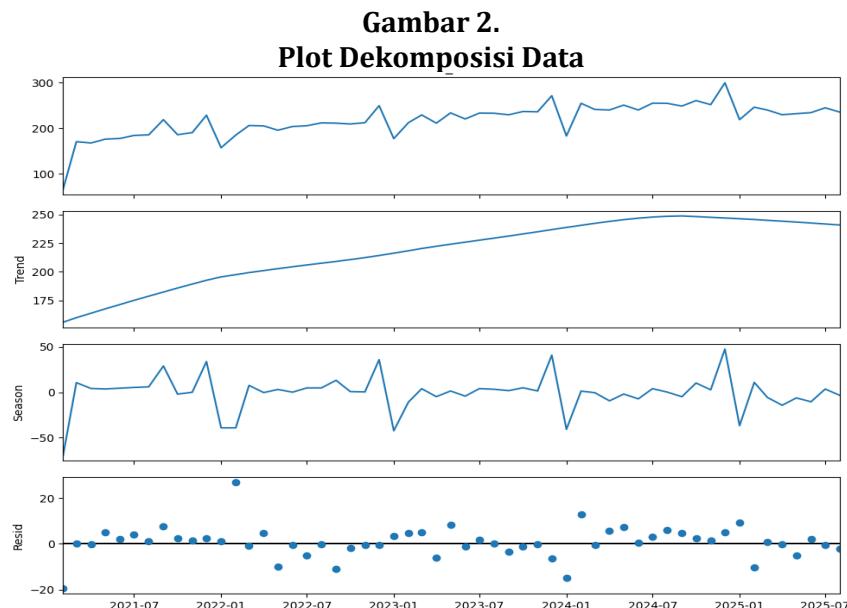
Selanjutnya, visualisasi data total iuran JKP sejak Februari 2021 hingga Agustus 2025 disajikan pada Gambar 1. Visualisasi data digunakan untuk mendeteksi pola tren dan musiman. Gambar 1 menunjukkan bahwa nilai iuran mengalami peningkatan secara umum selama periode observasi, yang mencerminkan peningkatan partisipasi dan kepatuhan perusahaan dalam mengikuti program JKP sejak implementasinya melalui PP No. 37 Tahun 2021. Nilai yang sangat rendah pada awal observasi (Februari 2021) dapat dijelaskan sebagai akibat dari tahap awal pelaksanaan program. Selain menunjukkan tren yang meningkat, grafik ini juga memperlihatkan fluktuasi nilai iuran dari waktu ke waktu, termasuk penurunan tajam yang berulang di awal tahun yang kemungkinan disebabkan oleh keterlambatan pelaporan atau faktor musiman administratif.

**Gambar 1.**  
**Iuran JKP Februari 2021-Agustus 2025**



Sumber: Diolah dari data primer

Selanjutnya, pada periode 2023 hingga 2024, nilai iuran cenderung lebih stabil, menandakan bahwa program mulai memasuki fase kematangan dan penerimaan iuran sudah berlangsung secara konsisten. Namun pada tahun 2025, terdapat pola yang berbeda di mana tren data menunjukkan penurunan yang cukup signifikan. Penurunan ini berkaitan erat dengan implementasi aturan iuran JKP yang baru, yaitu kebijakan mengenai penghentian atau pengurangan rasio iuran yang bersumber dari rekomposisi iuran Jaminan Kematian (JKM). Perubahan struktur pendanaan ini menyebabkan nilai nominal iuran JKP yang tercatat mengalami penurunan dibandingkan periode sebelumnya. Secara keseluruhan, grafik ini menunjukkan adanya pola tren dan musiman pada data, sehingga analisis lebih lanjut melalui teknik dekomposisi diperlukan untuk mengidentifikasi komponen-komponen tersebut secara eksplisit. Berikut ini disajikan visualisasi hasil dekomposisi data iuran JKP.



Sumber: Diolah dari data primer

Gambar 2 menunjukkan hasil dekomposisi data, yaitu *plot* dari pola tren, musiman, dan residual dari data. Pada plot *Trend*, dapat dilihat bahwa penerimaan iuran JKP mengalami kenaikan hingga akhir tahun 2024 dan mengalami penurunan pada tahun 2025. Kondisi ini disebabkan oleh dikeluarkannya kebijakan terbaru mengenai pengurangan *rate* iuran yang bersumber dari rekomposisi (JKM). Selanjutnya, pada plot *Season* terlihat adanya pola musiman yang berulang dalam data, khususnya terlihat jelas pada bulan Desember dan Januari. Pola ini mengindikasikan adanya pengaruh faktor musiman yang terjadi secara periodik, meskipun tidak selalu dengan intensitas yang sama setiap tahunnya. Pada plot *Residual* di baris keempat, sebagian besar titik tersebar di sekitar garis nol, yang menunjukkan bahwa variasi data yang tidak dijelaskan oleh tren dan musiman bersifat acak. Namun demikian, terdapat beberapa titik

*outlier* yang mengindikasikan adanya kejadian khusus atau anomali pada waktu tertentu yang tidak dapat dijelaskan oleh pola utama dalam data.

Temuan adanya tren, pola musiman, serta residual yang mengandung variasi acak dan outlier menunjukkan bahwa karakteristik data iuran JKP tidak sepenuhnya dapat dijelaskan oleh pendekatan linier. Kondisi ini memperkuat relevansi penggunaan pendekatan *hybrid* yang mengombinasikan model statistik dan *machine learning*.

Untuk menganalisis karakteristik statistik deret waktu, dilakukan pengujian terhadap kestasioneran data. Uji kestasioneran dilakukan menggunakan Uji *Augmented Dickey-Fuller* (ADF). Uji ADF digunakan untuk menguji hipotesis nol bahwa data memiliki akar unit atau tidak *stasioner*. Berikut hasil dari uji ADF terhadap data penerimaan iuran JKP:

<b>Tabel 1.</b> <b>Hasil Uji ADF</b>	
Statistik ADF	-2,649
p-value	0,083
Nilai Kritis 1%	-3,597
Nilai Kritis 5%	-2,933
Nilai Kritis 10%	-2,604

Sumber: Diolah dari data primer

Berdasarkan Tabel 1, diperoleh nilai uji *Augmented Dickey-Fuller* (ADF) sebesar -2,649 dengan *p-value* 0,083 ( $> \alpha = 0,05$ ). Dengan demikian, dapat disimpulkan bahwa hipotesis nol gagal ditolak, yang menunjukkan bahwa data memiliki akar unit (tidak stasioner) pada tingkat signifikansi 5%. Ketidakstasioneran data ini mengindikasikan bahwa model peramalan perlu memiliki kemampuan adaptif terhadap perubahan tren dan struktur data dari waktu ke waktu, sehingga pendekatan yang fleksibel seperti *Prophet* dan BiLSTM menjadi lebih sesuai dibandingkan model deret waktu konvensional. Data masih mengandung tren sehingga perlu dilakukan tahap prapemrosesan data sebelum masuk ke tahap pemodelan lebih lanjut.

Pada tahap prapemrosesan, observasi pertama (bulan Februari 2021) dari data iuran JKP dihapus karena memiliki nilai yang jauh lebih rendah dibandingkan observasi lainnya. Selanjutnya, dilakukan dua tahapan, yaitu *data splitting* (pembagian data) dan *data normalization* (normalisasi data).

*Data splitting* dilakukan dengan membagi data menjadi data latih (*training set*) dan data uji (*testing set*). Data pada penelitian ini akan dibagi menjadi 42 periode observasi pertama (Maret 2021 hingga Agustus 2024) sebagai data latih dan 12 periode observasi berikutnya (September 2024 hingga Agustus 2025) sebagai data uji. Tidak ada proses pengacakan dalam pembagian data ini untuk menjaga sifat data runtun waktu.

Sebelum melakukan pembentukan model *Bidirectional LSTM*, data dinormalisasi menggunakan metode *min-max normalization* dengan rentang [0,1]. Normalisasi ini diperlukan karena model LSTM dan turunannya sensitif terhadap perbedaan skala data, sehingga transformasi ini membantu meningkatkan stabilitas dan efektivitas proses pelatihan model.

Agar data runtun waktu dapat digunakan dalam kerangka *supervised learning*, diterapkan teknik *window generator*. Pada penelitian ini, ukuran jendela (*window size*) ditetapkan sebesar 12 periode. Artinya, model menggunakan 12 observasi historis berturut-turut sebagai input untuk memprediksi satu nilai pada periode berikutnya. Penggunaan window 12 memungkinkan model belajar mengenali pola pergerakan data dalam satu tahun sebelumnya untuk memperkirakan nilai pada periode selanjutnya.

Secara matematis, jika sebuah data runtun waktu direpresentasikan sebagai  $D = \{d_1, d_2, \dots, d_T\}$ , maka *window generator* akan menghasilkan sampel pertama dengan *input*  $X_1 = \{d_1, d_2, \dots, d_{12}\}$  dan label  $y_1 = d_{13}$ . Proses ini berlanjut dengan menggeser jendela satu langkah ke depan (*stride=1*), sehingga sampel ke- $i$  memiliki bentuk  $X_i = \{d_i, d_{i+1}, \dots, d_{i+11}\}$  yang berkorespondensi dengan label  $y_i = d_{i+12}$ . Transformasi ini menghasilkan struktur input yang sesuai untuk model BiLSTM, sehingga model dapat menganalisis keterkaitan antar waktu secara dua arah (maju dan mundur) dalam rentang 12 periode, yang penting untuk menangkap pola temporal yang kompleks.

### Pemodelan Menggunakan *Prophet*

Pemodelan menggunakan algoritma *Prophet* dilakukan melalui dua tahapan utama, yaitu pembentukan model awal (*baseline*) dengan konfigurasi *hyperparameter default* dan tahap optimasi model melalui proses *hyperparameter tuning*. Tahapan ini dilakukan guna memperoleh konfigurasi parameter yang menghasilkan performa prediksi terbaik dalam meramalkan nilai iuran JKP.

Model *baseline Prophet* dikonstruksi menggunakan konfigurasi *hyperparameter* bawaan tanpa penyesuaian khusus. Model dilatih menggunakan data latih yang telah diproses sebelumnya, kemudian digunakan untuk melakukan prediksi terhadap data uji. Evaluasi performa model dilakukan dengan membandingkan nilai prediksi terhadap nilai aktual menggunakan metrik *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan *Mean Absolute Percentage Error* (MAPE). Hasil evaluasi model awal disajikan pada Tabel 2.

**Tabel 2.**  
**Performa Model Baseline Prophet**

Model	Dataset	MSE	RMSE	MAE	MAPE
<i>Baseline</i>	Latih	$2,078 \times 10^{19}$	$4,559 \times 10^9$	$3,654 \times 10^9$	1,657
<i>Baseline</i>	Uji	$8,025 \times 10^{20}$	$2,833 \times 10^{10}$	$2,476 \times 10^{10}$	10,376

Sumber: Diolah dari data primer

Berdasarkan hasil evaluasi pada Tabel 2, terlihat bahwa model *baseline Prophet* mampu memberikan estimasi awal yang cukup baik, khususnya pada data latih. Akan tetapi tingginya nilai RMSE dan MAPE pada data uji menunjukkan bahwa akurasi model cenderung menurun saat dihadapkan pada data di luar periode pelatihan. Hal ini menunjukkan adanya potensi peningkatan performa melalui optimasi parameter model.

Untuk meningkatkan performa model *baseline*, dilakukan proses *hyperparameter tuning* menggunakan pendekatan *grid search*. Proses ini bertujuan mengeksplorasi berbagai kombinasi nilai *hyperparameter* yang telah ditentukan sebelumnya guna menemukan konfigurasi yang paling optimal. Daftar kandidat nilai *hyperparameter* untuk proses *tuning* disajikan pada Tabel 3.

**Tabel 3**  
**Konfigurasi Nilai Hyperparameter untuk Grid Search**

Hyperparameter	Kandidat Nilai
<i>changepoint_prior_scale</i>	{0,01, 0,05, 0,1, 0,15, 0,2}
<i>seasonality_prior_scale</i>	{0,01, 0,05, 0,1, 0,15, 0,2}
<i>seasonality_mode</i>	{'additive', 'multiplicative'}

Sumber: Diolah dari data primer

Proses *tuning* menghasilkan 50 kombinasi model yang berbeda. Masing-masing model dievaluasi berdasarkan performanya dalam memprediksi data uji. Kombinasi *hyperparameter* terbaik dipilih berdasarkan hasil evaluasi yang paling rendah pada metrik MAPE. Konfigurasi parameter terbaik ditunjukkan pada Tabel 4, sementara hasil evaluasi model optimal disajikan pada Tabel 5.

**Tabel 4.**  
**Konfigurasi Hyperparameter Terbaik Hasil Grid Search**

Hyperparameter	Nilai Terpilih
<i>changepoint_prior_scale</i>	0,15
<i>seasonality_prior_scale</i>	0,01
<i>seasonality_mode</i>	'additive'

Sumber: Diolah dari data primer

Konfigurasi *hyperparameter* terbaik menunjukkan bahwa model *Prophet* menggunakan fleksibilitas tren yang moderat (*changepoint\_prior\_scale* = 0,15) sehingga mampu menangkap perubahan tren tanpa terlalu sensitif terhadap fluktuasi jangka pendek. Nilai *seasonality\_prior\_scale* yang kecil membatasi pengaruh musiman agar tidak mendominasi model, sementara pemilihan *seasonality\_mode* aditif menunjukkan bahwa pola musiman bersifat relatif stabil dan ditambahkan secara langsung pada komponen tren. Kombinasi ini menghasilkan model yang seimbang antara akurasi dan kemampuan generalisasi.

**Tabel 5**  
**Metrik Evaluasi Model *Prophet* Hasil Tuning**

<b>Model</b>	<b>Dataset</b>	<b>MSE</b>	<b>RMSE</b>	<b>MAE</b>	<b>MAPE</b>
<i>Tuned</i>	Latih	$3,889 \times 10^{19}$	$6,236 \times 10^9$	$4,851 \times 10^9$	2,205
<i>Tuned</i>	Uji	$5,937 \times 10^{20}$	$2,437 \times 10^{10}$	$2,082 \times 10^{10}$	8,685

Sumber: Diolah dari data primer

Dengan membandingkan hasil evaluasi model *baseline* dan model hasil *tuning*, dapat disimpulkan bahwa proses optimasi *hyperparameter* berhasil meningkatkan performa model *Prophet*, khususnya pada data uji. Penurunan nilai RMSE dari  $2,83 \times 10^{10}$  menjadi  $2,43 \times 10^{10}$  serta penurunan MAPE dari 10,37% menjadi 8,68% menunjukkan peningkatan akurasi prediksi model terhadap nilai aktual. Temuan ini mengindikasikan bahwa pemilihan konfigurasi parameter yang lebih tepat memungkinkan model *Prophet* menangkap perubahan tren dan pola musiman secara lebih optimal, sehingga menghasilkan kemampuan generalisasi yang lebih baik dalam konteks peramalan penerimaan iuran JKP.

### Pemodelan Menggunakan *Bidirectional LSTM*

Pemodelan deret waktu menggunakan *Bidirectional Long Short-Term Memory* (BiLSTM) dilakukan dalam dua tahapan. Tahapan pertama adalah pembangunan model awal (*baseline*) menggunakan konfigurasi *hyperparameter* standar, sedangkan tahapan kedua mencakup proses *hyperparameter tuning* untuk mengoptimalkan performa model. Tujuan dari proses ini adalah untuk meningkatkan akurasi prediksi iuran JKP dengan memanfaatkan kemampuan BiLSTM dalam menangkap pola temporal kompleks dan hubungan jangka panjang dalam data.

Model awal dikembangkan menggunakan arsitektur BiLSTM dengan konfigurasi dasar, yaitu jumlah unit tersembunyi sebanyak 64, fungsi aktivasi *TanH*, dan jumlah *epoch* sebanyak 50. Model dilatih menggunakan data latih, kemudian dievaluasi pada data uji untuk mengukur kemampuan generalisasi model.

Evaluasi dilakukan menggunakan empat metrik utama, yaitu *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan *Mean Absolute Percentage Error* (MAPE). Ringkasan performa model awal disajikan pada Tabel 6.

**Tabel 6.**  
**Performa Model Baseline BiLSTM**

Model	Dataset	MSE	RMSE	MAE	MAPE
<i>Baseline</i>	Latih	$1,126 \times 10^{20}$	$1,061 \times 10^{10}$	$8,217 \times 10^9$	5,031
<i>Baseline</i>	Uji	$2,287 \times 10^{21}$	$4,783 \times 10^{10}$	$4,070 \times 10^{10}$	17,054

Sumber: Diolah dari data primer

Berdasarkan hasil evaluasi, model *baseline* BiLSTM menunjukkan kemampuan awal yang cukup baik dalam memodelkan data deret waktu iuran JKP pada data latih, yang ditunjukkan oleh nilai MAPE sebesar 5,031%. Namun, nilai MAPE yang relatif tinggi pada data uji sebesar 17,054% mengindikasikan bahwa model *baseline* belum mampu melakukan generalisasi dengan baik terhadap data di luar data latih.

Selain itu, tingginya nilai RMSE dan MSE pada kedua dataset menunjukkan bahwa model masih perlu dioptimalkan untuk memperoleh prediksi yang lebih akurat dan stabil. Untuk meningkatkan performa model awal, dilakukan proses *hyperparameter tuning* menggunakan metode *grid search*. Metode ini mengevaluasi berbagai kombinasi parameter untuk menemukan konfigurasi terbaik yang mampu meningkatkan akurasi model. Kandidat nilai *hyperparameter* yang diuji ditampilkan pada Tabel 7, dengan total 18 kombinasi.

**Tabel 7**  
**Konfigurasi Nilai Hyperparameter untuk Grid Search BiLSTM**

Hyperparameter	Kandidat Nilai
<i>units</i>	{64, 128, 256}
<i>epochs</i>	{50, 75, 100}
<i>activation</i>	{'relu', 'tanh'}

Sumber: Diolah dari data primer

Hasil dari proses *grid search* menunjukkan bahwa konfigurasi optimal diperoleh dengan menggunakan 64 unit LSTM, fungsi aktivasi *relu*, dan 100 *epoch* pelatihan. Konfigurasi terbaik tersebut ditunjukkan pada Tabel 8.

**Tabel 8.**  
**Konfigurasi *Hyperparameter* Terbaik Hasil *Grid Search* BiLSTM**

<i>Hyperparameter</i>	Nilai Terbaik
<i>units</i>	64
<i>epochs</i>	100
<i>activation</i>	'relu'

Sumber: Diolah dari data primer

Performa model BiLSTM hasil tuning ditampilkan pada Tabel 9. Dibandingkan dengan model *baseline*, model yang telah dioptimasi menunjukkan peningkatan performa secara signifikan.

**Tabel 9**  
**Metrik Evaluasi Model BiLSTM Hasil Tuning**

Model	Dataset	MSE	RMSE	MAE	MAPE
<i>Tuned</i>	Latih	$4,133 \times 10^{19}$	$6,429 \times 10^9$	$5,121 \times 10^9$	3,489
<i>Tuned</i>	Uji	$5,341 \times 10^{20}$	$2,244 \times 10^{10}$	$1,804 \times 10^{10}$	7,530

Sumber: Diolah dari data primer

Dari Tabel 9 dapat dilihat bahwa nilai RMSE pada data uji menurun dari  $4,783 \times 10^{10}$  menjadi  $2,244 \times 10^{10}$ , sedangkan nilai MAPE juga mengalami penurunan dari 17,05% menjadi 7,53%. Penurunan ini menunjukkan bahwa model BiLSTM yang telah dioptimasi memiliki performa prediktif yang lebih baik dalam menangkap pola deret waktu dan meminimalkan kesalahan prediksi.

### Pemodelan *Hybrid Prophet-BiLSTM*

Setelah memperoleh model *Prophet* dan BiLSTM terbaik melalui proses *hyperparameter tuning*, tahap selanjutnya dalam penelitian ini adalah membentuk model *hybrid* yang menggabungkan kedua model tersebut. Tujuan dari pendekatan ini adalah untuk memanfaatkan kelebihan masing-masing model dalam menangkap pola tren, musiman, dan komponen nonlinier yang terdapat pada data.

Model *hybrid* dibangun dengan mengombinasikan hasil prediksi dari model *Prophet* dan BiLSTM, di mana bobot kontribusi masing-masing model terhadap prediksi akhir ditentukan menggunakan metode LOWE (*Linear Optimal Weighting Estimator*). LOWE merupakan pendekatan non-iteratif yang berbasis regresi linier tanpa intersep yang bertujuan untuk memperoleh kombinasi linier optimal dari dua atau lebih model peramalan dengan meminimalkan selisih kuadrat antara prediksi gabungan dan nilai aktual.

Dalam implementasinya, diberikan bobot  $\alpha$  untuk model *Prophet* dan bobot  $\beta$  untuk model BiLSTM. Bobot  $\alpha$  dan  $\beta$  diperoleh melalui penyelesaian sistem persamaan *least squares*, dengan persyaratan  $\alpha + \beta = 1$ . Setelah dilakukan pembobotan menggunakan LOWE, didapatkan bobot optimal sebagai berikut:

$$\alpha = 0,9308$$

$$\beta = 0,0692$$

Setelah didapatkan bobot optimal, kemudian model *hybrid* ini diuji pada data uji dan menghasilkan performa pada Tabel 10 berikut:

**Tabel 10**  
**Metrik Evaluasi Model *Hybrid Prophet-BiLSTM***

Dataset	MSE	RMSE	MAE	MAPE
Latih	$4,372 \times 10^{19}$	$6,612 \times 10^9$	$5,642 \times 10^9$	2,521
Uji	$5,394 \times 10^{20}$	$2,322 \times 10^{10}$	$1,953 \times 10^{10}$	8,129

Sumber: Diolah dari data primer

Berdasarkan Tabel 10 dan Tabel 11 dapat dilihat bahwa model *hybrid* memiliki performa yang lebih baik dibandingkan model *Prophet*, namun belum melampaui performa model BiLSTM pada data uji. Berikut merupakan perbandingan prediksi dari model *hybrid* untuk data uji dan data latih.

**Tabel 11**  
**Perbandingan Metrik Evaluasi Model *Prophet*, *BiLSTM*, dan *Hybrid***

Model	Dataset	MSE	RMSE	MAE	MAPE
<i>Prophet</i>	Latih	$3,889 \times 10^{19}$	$6,236 \times 10^9$	$4,851 \times 10^9$	2,205
BiLSTM	Latih	$4,133 \times 10^{19}$	$6,429 \times 10^9$	$5,121 \times 10^9$	3,489
<i>Hybrid</i>	Latih	$4,372 \times 10^{19}$	$6,612 \times 10^9$	$5,642 \times 10^9$	2,521
<i>Prophet</i>	Uji	$5,937 \times 10^{20}$	$2,437 \times 10^{10}$	$2,082 \times 10^{10}$	8,685
BiLSTM	Uji	$5,341 \times 10^{20}$	$2,244 \times 10^{10}$	$1,804 \times 10^{10}$	7,530
<i>Hybrid</i>	Uji	$5,394 \times 10^{20}$	$2,322 \times 10^{10}$	$1,953 \times 10^{10}$	8,129

Sumber: Diolah dari data primer

Tabel 11 menyajikan perbandingan kinerja tiga model peramalan, yaitu *Prophet*, BiLSTM, dan model *Hybrid* (gabungan *Prophet* dan BiLSTM), berdasarkan metrik evaluasi MSE, RMSE, MAE, dan MAPE pada data latih dan data uji.

---

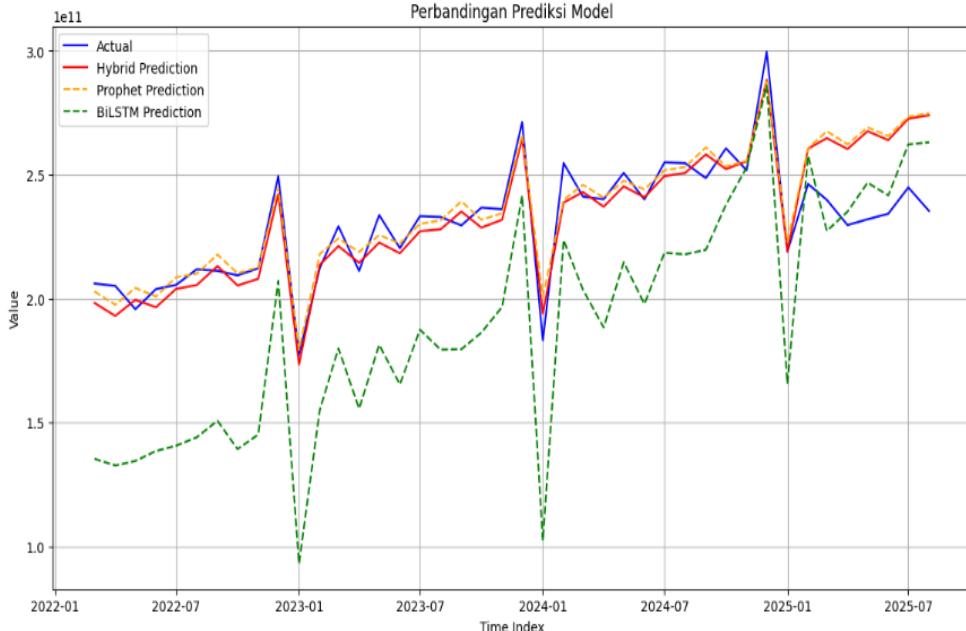
Pada data latih, model *Prophet* menunjukkan performa terbaik dibandingkan dua model lainnya, dengan nilai MSE, RMSE, MAE, dan MAPE yang paling rendah. Temuan ini mengindikasikan bahwa model *Prophet* sangat efektif dalam mempelajari pola tren dan musiman yang terkandung dalam data historis. Sementara itu, model BiLSTM memiliki nilai *error* yang relatif lebih tinggi pada fase pelatihan, yang mengindikasikan model belum mencapai pembelajaran yang stabil pada data latih dan berpotensi mengalami kecenderungan *overtraining* ketika dihadapkan pada variasi data yang berbeda. Model *hybrid* berada pada posisi menengah, yang mencerminkan bahwa penggabungan dua pendekatan menghasilkan proses pembelajaran yang lebih seimbang, tidak terlalu “mengikuti” data latih seperti *Prophet*, tetapi juga tidak menunjukkan ketidakstabilan seperti BiLSTM pada tahap pelatihan.

Berbeda dengan hasil pada data latih, evaluasi pada data uji menunjukkan adanya pergeseran performa. Model *Prophet* mengalami kenaikan *error* yang paling jelas dari data latih ke data uji, yang menandakan keterbatasan dalam generalisasi meskipun sangat baik dalam memodelkan tren dan musiman pada data historis. Di sisi lain, BiLSTM memang menghasilkan *error* yang lebih rendah pada data uji, namun pola perbedaan yang tajam antara performa latih dan uji mengindikasikan adanya risiko *overtraining*, yakni model dapat sangat sensitif terhadap pola tertentu sehingga performanya tidak selalu konsisten ketika pola data berubah. Dalam konteks ini, model *hybrid* menunjukkan performa yang lebih stabil dan seimbang, karena mampu menjaga *error* tetap kompetitif pada data uji sekaligus tidak terlalu bergantung pada kecocokan yang tinggi di data latih.

Secara keseluruhan, dalam studi ini model *hybrid* dinilai paling unggul karena mampu mengombinasikan kekuatan model *Prophet* dalam menangkap tren dan musiman serta keunggulan BiLSTM dalam mengenali pola nonlinier jangka pendek, sehingga menghasilkan prediksi yang lebih akurat dan seimbang, khususnya dalam konteks penerapan pada data baru. Dengan demikian, meskipun model *hybrid* tidak selalu menghasilkan nilai *error* paling rendah, kestabilan kinerja dan kemampuan dalam menggabungkan kelebihan kedua model menjadikan pendekatan *hybrid* lebih reliabel untuk kebutuhan peramalan. Gambar 3 berikut merupakan gambar prediksi antar model *hybrid* dengan model individual pada data latih dan data uji.

Gambar 3 menunjukkan bahwa model *Prophet* dan model *hybrid* mampu mengikuti pola aktual data iuran JKP dengan cukup baik, terutama dalam menangkap tren meningkat dan fluktuasi musiman utama. Model BiLSTM terlihat lebih fluktuatif dan cenderung menyimpang pada beberapa periode, khususnya saat terjadi perubahan nilai yang tajam, yang mengindikasikan sensitivitas tinggi terhadap variasi data.

**Gambar 3.**  
**Perbandingan Prediksi Antar Model**



Sumber: Diolah dari data primer

Sementara itu, model *hybrid* menghasilkan garis prediksi yang lebih halus dan stabil dibandingkan BiLSTM, serta lebih adaptif dibandingkan *Prophet* pada periode tertentu, sehingga mencerminkan kemampuan kombinasi kedua model dalam menghasilkan prediksi yang seimbang dan relatif konsisten terhadap data aktual.

### Validasi Model

Setelah proses pemodelan diselesaikan, tahap selanjutnya adalah melakukan validasi guna mengevaluasi kinerja dan tingkat *robustness* dari masing-masing model yang dikembangkan. Validasi ini dilakukan untuk memastikan bahwa performa model tidak hanya optimal pada data latih, tetapi juga konsisten dan dapat diandalkan ketika dihadapkan pada data yang belum pernah digunakan sebelumnya (*out-of-sample*). Penggunaan satu pasangan data latih dan uji secara tunggal berisiko menghasilkan estimasi performa yang bias dan tidak representatif terhadap variasi data sepanjang waktu.

Untuk mengatasi keterbatasan tersebut, penelitian ini menerapkan pendekatan validasi silang yang dirancang khusus untuk data runtun waktu, yaitu metode *rolling window*. Metode ini dipilih karena mampu mempertahankan urutan temporal data. Dalam implementasinya, data dibagi menjadi lima segmen yang masing-masing disebut sebagai *window*. Pada setiap *window*, model dilatih menggunakan subset data historis, kemudian dievaluasi pada data yang

secara kronologis berada setelahnya dan tidak pernah digunakan dalam pelatihan. Hasil validasi performa model dari kelima *window* tersebut disajikan pada Tabel 12 dan Gambar 4.

**Tabel 12.**  
**Perbandingan Metrik Evaluasi Model Berdasarkan Window**

Window	Model	MSE	RMSE	MAE	MAPE
Window 1	<i>Prophet (Tuned)</i>	$4,123 \times 10^{19}$	$6,421 \times 10^9$	$6,320 \times 10^9$	2,541
Window 1	<i>BiLSTM (Tuned)</i>	$5,551 \times 10^{19}$	$7,451 \times 10^9$	$7,108 \times 10^9$	2,823
Window 1	<i>Hybrid</i>	$3,725 \times 10^{19}$	$6,103 \times 10^9$	$6,086 \times 10^9$	2,439
Window 2	<i>Prophet (Tuned)</i>	$3,277 \times 10^{20}$	$1,810 \times 10^{10}$	$1,445 \times 10^{10}$	5,724
Window 2	<i>BiLSTM (Tuned)</i>	$2,166 \times 10^{19}$	$4,654 \times 10^9$	$4,121 \times 10^9$	1,621
Window 2	<i>Hybrid</i>	$2,037 \times 10^{20}$	$1,427 \times 10^{10}$	$1,197 \times 10^{10}$	4,739
Window 3	<i>Prophet (Tuned)</i>	$4,798 \times 10^{19}$	$6,927 \times 10^9$	$6,797 \times 10^9$	2,755
Window 3	<i>BiLSTM (Tuned)</i>	$4,053 \times 10^{20}$	$2,013 \times 10^{10}$	$1,737 \times 10^{10}$	6,808
Window 3	<i>Hybrid</i>	$1,614 \times 10^{20}$	$1,270 \times 10^{10}$	$1,163 \times 10^{10}$	4,609
Window 4	<i>Prophet (Tuned)</i>	$9,845 \times 10^{20}$	$3,138 \times 10^{10}$	$3,108 \times 10^{10}$	13,318
Window 4	<i>BiLSTM (Tuned)</i>	$4,163 \times 10^{20}$	$2,040 \times 10^{10}$	$1,929 \times 10^{10}$	8,298
Window 4	<i>Hybrid</i>	$7,670 \times 10^{20}$	$2,769 \times 10^{10}$	$2,724 \times 10^{10}$	11,680
Window 5	<i>Prophet (Tuned)</i>	$1,549 \times 10^{20}$	$1,244 \times 10^{10}$	$1,180 \times 10^{10}$	4,977
Window 5	<i>BiLSTM (Tuned)</i>	$1,714 \times 10^{19}$	$4,140 \times 10^9$	$3,958 \times 10^9$	1,660
Window 5	<i>Hybrid</i>	$9,187 \times 10^{19}$	$9,585 \times 10^9$	$8,768 \times 10^9$	3,706

Sumber: Diolah dari data primer

Tabel 12 menyajikan nilai metrik *error* dari masing-masing model, dengan rincian sebagai berikut:

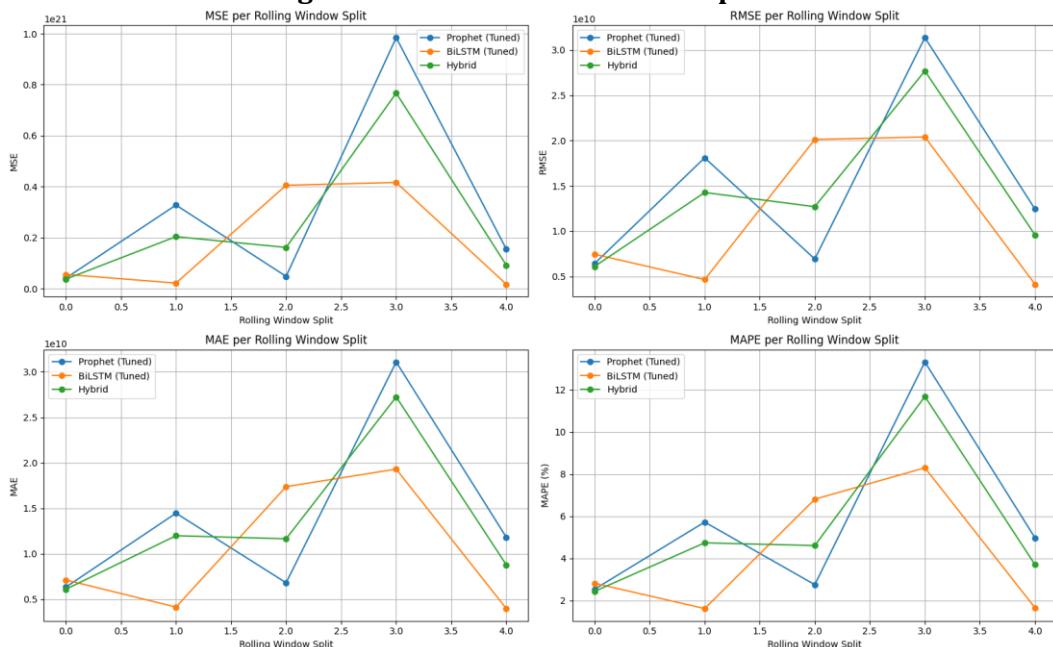
- Pada *Window 1*, sebanyak 39 data digunakan sebagai data latih yang mencakup periode Maret 2021 hingga Mei 2024. Kemudian, 3 data terakhir, yaitu dari Juni 2024 hingga Agustus 2024, digunakan sebagai data uji.
- Pada *Window 2*, sebanyak 42 data digunakan sebagai data latih yang mencakup periode Maret 2021 hingga Agustus 2024. Kemudian, 3 data terakhir, yaitu dari September 2024 hingga November 2024, digunakan sebagai data uji.
- Pada *Window 3*, sebanyak 45 data digunakan sebagai data latih yang mencakup periode Maret 2021 hingga November 2024. Kemudian, 3 data terakhir, yaitu dari Desember 2024 hingga Februari 2025, digunakan sebagai data uji.

- Pada *Window 4*, sebanyak 48 data digunakan sebagai data latih yang mencakup periode Maret 2021 hingga Februari 2025. Kemudian, 3 data terakhir, yaitu dari Maret 2025 hingga Mei 2025, digunakan sebagai data uji.
- Pada *Window 5*, sebanyak 51 data digunakan sebagai data latih yang mencakup periode Maret 2021 hingga Mei 2025. Kemudian, 3 data terakhir, yaitu dari Juni 2025 hingga Agustus 2025, digunakan sebagai data uji.

Berdasarkan Tabel 12, model *Prophet* dan BiLSTM menunjukkan performa yang bervariasi pada lima periode evaluasi yang digunakan. Tidak ada satu model pun yang secara konsisten menunjukkan performa terbaik di seluruh metrik. Model *Prophet* cenderung memiliki nilai *error* yang lebih tinggi, terutama pada *Window 2* dan *Window 4*, sedangkan BiLSTM menunjukkan kinerja yang relatif stabil dengan keunggulan pada beberapa *window*.

Sebagai contoh, pada *Window 2*, *4*, dan *5*, BiLSTM mencatatkan nilai RMSE, MAE, dan MAPE yang lebih rendah dibandingkan model *Prophet* maupun *hybrid*. Sementara itu, model *Hybrid Prophet–BiLSTM* menunjukkan keunggulan yang menonjol pada *Window 1*, di mana seluruh metrik *error* (MSE, RMSE, MAE, dan MAPE) tercatat paling rendah dibandingkan kedua model lainnya. Namun, pada *window* lainnya, performa model *hybrid* tidak selalu lebih baik dibandingkan BiLSTM. Temuan ini menunjukkan bahwa tidak terdapat satu model yang secara konsisten unggul di semua periode evaluasi, meskipun BiLSTM cenderung memberikan performa terbaik secara keseluruhan.

**Gambar 4.**  
**Perbandingan Performa Model Pada Setiap Metrik Error**



Sumber: Diolah dari data primer

Gambar 4 merupakan gambaran visual dari Tabel 12. Garis berwarna hijau yang mewakili model *hybrid* cenderung berada pada rentang *error* yang lebih stabil dan tidak menunjukkan lonjakan ekstrem dibandingkan model lainnya.

Ketidakstabilan nilai metrik *error* pada beberapa *Window* belum diketahui secara pasti penyebabnya karena keterbatasan waktu dalam analisis lebih lanjut. Namun, berdasarkan hasil pada Gambar 4 serta Tabel 12, terlihat bahwa seluruh model mengalami peningkatan performa pada *Window 4*. Hal ini diduga berkaitan dengan karakteristik data pada periode pelatihan dan pengujian di *Window 4* yang cenderung lebih stabil atau memiliki pola yang lebih mudah dipelajari oleh model. Analisis mendalam terhadap pola musiman, tren, atau adanya *outlier* pada periode tersebut diperlukan untuk memastikan penyebab penurunan *error* secara menyeluruh.

Berdasarkan hasil validasi dengan metode *rolling window* sebelumnya, model *Hybrid Prophet-BiLSTM* akan digunakan untuk melakukan peramalan pada tahap selanjutnya karena model ini menunjukkan kinerja yang relatif stabil di berbagai *window* evaluasi, tanpa fluktuasi *error* yang ekstrem, serta mampu menggabungkan keunggulan *Prophet* dan BiLSTM dalam menghadapi dinamika data dari waktu ke waktu.

### Peramalan Iuran Jaminan Kehilangan Pekerjaan

Peramalan dilakukan menggunakan model *Hybrid Prophet-BiLSTM* dengan bobot untuk model *Prophet* adalah 0,9308 dan bobot untuk model BiLSTM adalah 0,0692. Model ini digunakan untuk memproyeksikan nilai iuran JKP pada periode satu tahun setelah data observasi terakhir, yaitu dari September 2025 hingga Agustus 2026. Hasil peramalan disajikan dalam bentuk titik prediksi serta visualisasi grafik untuk memudahkan interpretasi pola proyeksi.

**Tabel 13.**  
**Tabel Hasil Peramalan JKP Menggunakan Model *Hybrid Prophet-BiLSTM***

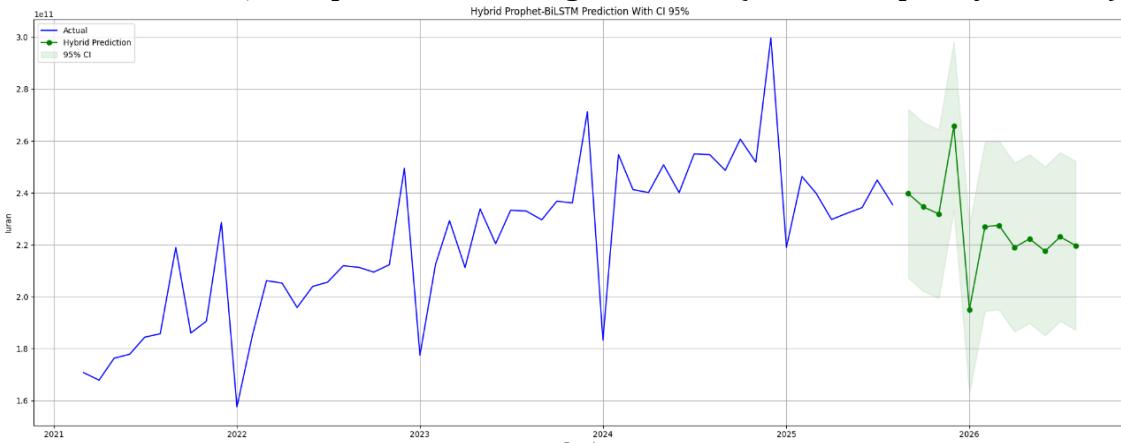
Tanggal	Hybrid	Hybrid_lower	Hybrid_upper
01-08-2024	$239,734 \times 10^9$	$207,208 \times 10^9$	$272,260 \times 10^9$
01-09-2024	$234,635 \times 10^9$	$202,109 \times 10^9$	$267,161 \times 10^9$
01-10-2024	$231,855 \times 10^9$	$199,329 \times 10^9$	$264,381 \times 10^9$
01-11-2024	$265,676 \times 10^9$	$233,150 \times 10^9$	$298,203 \times 10^9$
01-12-2024	$195,017 \times 10^9$	$162,491 \times 10^9$	$227,543 \times 10^9$
01-01-2025	$227,013 \times 10^9$	$194,487 \times 10^9$	$259,539 \times 10^9$
01-02-2025	$227,484 \times 10^9$	$194,958 \times 10^9$	$260,010 \times 10^9$
01-03-2025	$218,989 \times 10^9$	$186,463 \times 10^9$	$251,515 \times 10^9$

Tanggal	<i>Hybrid</i>	<i>Hybrid_lower</i>	<i>Hybrid_upper</i>
01-04-2025	$222,300 \times 10^9$	$189,774 \times 10^9$	$254,826 \times 10^9$
01-05-2025	$217,546 \times 10^9$	$185,020 \times 10^9$	$250,072 \times 10^9$
01-06-2025	$223,052 \times 10^9$	$190,526 \times 10^9$	$255,578 \times 10^9$
01-07-2025	$219,702 \times 10^9$	$187,176 \times 10^9$	$252,228 \times 10^9$

Sumber: Diolah dari data primer

Berdasarkan hasil peramalan yang ditampilkan pada Tabel 13, nilai iuran JKP periode September 2025 hingga Agustus 2026 diproyeksikan berada dalam kisaran Rp195,02 miliar hingga Rp265,68 miliar. Nilai tertinggi diperkirakan terjadi pada Desember 2025 sebesar Rp265,68 miliar, sedangkan nilai terendah diperkirakan terjadi pada Januari 2026 sebesar Rp195,02 miliar. Pola ini menunjukkan adanya fluktuasi musiman yang konsisten dengan karakteristik historis data, di mana iuran cenderung meningkat menjelang akhir tahun dan mengalami penurunan pada awal tahun.

**Gambar 5.**  
**Peramalan Iuran JKP September 2025 - Agustus 2026 (Interval Kepercayaan 95%)**



Sumber: Diolah dari data primer

Selain menghasilkan nilai titik prediksi, setiap estimasi dilengkapi dengan interval kepercayaan 95% sebagaimana ditunjukkan pada Gambar 5. Interval kepercayaan ini merepresentasikan tingkat ketidakpastian prediksi yang melekat pada proses peramalan. Sebagai contoh, pada September 2025, nilai prediksi iuran JKP tercatat sebesar Rp239,73 miliar, dengan batas bawah interval kepercayaan sebesar Rp207,21 miliar dan batas atas sebesar Rp272,26 miliar. Dengan demikian, pada tingkat keyakinan 95%, nilai aktual iuran JKP diperkirakan berada dalam rentang tersebut. Informasi ini memberikan konteks yang penting dalam menilai reliabilitas hasil peramalan dan mendukung pengambilan keputusan berbasis risiko.

Secara historis, data iuran JKP menunjukkan tren peningkatan sejak awal implementasi program. Namun, pada tahun 2025 terlihat adanya perlambatan bahkan penurunan tren, yang berkaitan dengan perubahan kebijakan terkait struktur dan besaran iuran. Perubahan ini terekam dalam komponen tren model, sehingga mempengaruhi arah proyeksi satu tahun ke depan. Akibatnya, hasil peramalan pada periode September 2025 hingga Agustus 2026 memperlihatkan pola yang relatif stabil dan tidak menunjukkan peningkatan yang signifikan, berbeda dengan pola proyeksi pada tahun-tahun sebelumnya yang cenderung naik secara konsisten. Temuan ini mengindikasikan bahwa model *hybrid* berhasil mengadaptasi perubahan dinamika tren terkini dalam proses estimasi.

Secara keseluruhan, hasil peramalan ini menunjukkan bahwa model *Hybrid Prophet-BiLSTM* mampu menangkap kombinasi pola tren jangka panjang dan fluktuasi musiman nonlinier dengan baik. Proyeksi yang dihasilkan dapat dimanfaatkan sebagai dasar dalam perencanaan keuangan dan pengelolaan dana JKP, termasuk dalam perumusan strategi antisipatif terhadap periode dengan nilai iuran yang rendah, seperti pada Januari 2026, serta pemanfaatan potensi surplus pada periode nilai iuran tinggi, seperti Desember 2025. Dengan demikian, hasil peramalan ini tidak hanya memberikan estimasi kuantitatif, tetapi juga mendukung pengambilan keputusan strategis yang adaptif terhadap perubahan tren dan dinamika musiman iuran JKP.

## Kesimpulan

Penelitian ini bertujuan untuk mengembangkan model peramalan iuran Jaminan Kehilangan Pekerjaan (JKP) yang lebih akurat dan stabil melalui pendekatan *Hybrid Prophet-BiLSTM* dengan pembobotan menggunakan metode *Linear Optimal Weighting Estimator* (LOWE). Berdasarkan proses evaluasi yang telah dilakukan, dapat disimpulkan bahwa pendekatan *hybrid* ini secara efektif mampu mengintegrasikan keunggulan masing-masing model, yaitu model *Prophet* dalam menangkap pola tren dan musiman, serta BiLSTM dalam mengenali dinamika nonlinier dan jangka panjang. Dominasi bobot *Prophet* (0.9308) dalam kombinasi akhir mencerminkan kekuatan struktur musiman yang konsisten pada data iuran JKP, sementara kontribusi BiLSTM turut memperkuat aspek fleksibilitas dan adaptabilitas terhadap pola yang kompleks.

Evaluasi menggunakan metrik *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan *Mean Absolute Percentage Error* (MAPE) menunjukkan bahwa model *Hybrid Prophet-BiLSTM* menghasilkan kinerja prediksi yang lebih baik dan lebih stabil dibandingkan model *Prophet* dan BiLSTM secara parsial, khususnya pada data uji dan skema validasi *rolling window*. Temuan ini mengonfirmasi hipotesis penelitian bahwa

penggabungan kedua model melalui pembobotan *Linear Optimal Weighting Estimator* (LOWE) mampu meningkatkan akurasi dan kemampuan generalisasi model peramalan.

Hasil ini mendukung hipotesis bahwa model *Hybrid Prophet-BiLSTM* yang dibobot secara optimal dapat menjadi alat prediktif yang andal bagi keperluan perencanaan strategis BPJS Ketenagakerjaan, khususnya dalam menghadapi dinamika tren dan fluktuasi musiman penerimaan iuran JKP.

## Referensi

- Almazrouee, Abdulla I., Abdullah M. Almeshal, Abdulrahman S. Almutairi, Mohammad R. Alenezi, dan Saleh N. Alhajeri. 2020. "Long-Term Forecasting of Electrical Loads in Kuwait Using Prophet and Holt-Winters Models." *Applied Sciences* 10(16):5627. doi: 10.3390/app10165627.
- Althelaya, Khaled A., El-Sayed M. El-Alfy, dan Salahadin Mohammed. 2018. "Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU)." Dalam *2018 21st Saudi Computer Society National Computer Conference (NCC)*, 1–7. doi: 10.1109/NCG.2018.8593076.
- Arslan, Sedar. 2022. "A hybrid forecasting model using LSTM and Prophet for energy consumption with decomposition of time series data." *PeerJ Computer Science* 8: e1001.
- Bates, Stephen, Trevor Hastie, dan Robert Tibshirani. 2024. "Cross-Validation: What Does It Estimate and How Well Does It Do It?" *Journal of the American Statistical Association* 119(546):1434–1445. doi: <https://doi.org/10.1080/01621459.2023.2197686>.
- Bergmeir, Christoph, dan José M. Benítez. 2012. "On the use of cross-validation for time series predictor evaluation." *Information Sciences* 191:192–213. doi: 10.1016/j.ins.2011.12.028.
- Cerqueira, Vítor, Luís Torgo, dan Igor Mozetič. 2020. "Evaluating forecasting performance with cross-validation." *International Journal of Forecasting* 36(2):413–431. doi: 10.1016/j.ijforecast.2019.05.011.
- Chahkotahi, Fatemeh dan Mehdi Khashei. 2022. "Linear Optimal Weighting Estimator (LOWE) for Efficient Parallel Hybridization of Load Forecasts." *Journal of Modelling in Management* 17(3):1028–1048. doi: <https://doi.org/10.1108/JM2-05-2021-0116>.
- Choubey, Mohit, Rahul Kumar Chaurasiya, dan J. S. Yadav. 2025. "Predicting Electrical Load Demands Using Neural Prophet-Based Forecasting Model." *SN Computer Science* 6(1):46. doi: 10.1007/s42979-024-03587-6.

- Claesen, Marc, dan Bart De Moor. 2015. "Hyperparameter Search in Machine Learning." *arXiv*. doi: 10.48550/arXiv.1502.02127.
- Ghojogh, Benyamin, dan Ali Ghodsi. 2023. "Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey." *arXiv*. <http://arxiv.org/abs/2304.11461>.
- Goodfellow, Ian, Yoshua Bengio, dan Aaron Courville. 2016. *Deep Learning*. Cambridge, MA: MIT Press.
- Kenyi, Manzu Gerald Simon dan Kayoko Yamamoto. 2024. "A hybrid SARIMA-Prophet model for predicting historical streamflow time-series of the Sobat River in South Sudan." *Discovered Applied Science* 6 (457). <https://doi.org/10.1007/s42452-024-06083-x>.
- Kementerian Ketenagakerjaan Republik Indonesia. 2024. *Kumpulan Data - Upah Minimum Kabupaten/Kota Menurut Provinsi*. Diakses 13 Februari 2025, dari situs resmi Satu Data Kemnaker.
- Kementerian Ketenagakerjaan Republik Indonesia. 2025. *Kumpulan Data - Data Angkatan Kerja, Penduduk Bekerja, dan Pengangguran Menurut Provinsi*. Diakses 13 Februari 2025, dari situs resmi Satu Data Kemnaker.
- Kwarteng, Samuel Baffoe dan Poguda Aleksey Andreevich. 2024. "Comparative Analysis of ARIMA, SARIMA and Prophet Model in Forecasting." *Research & Development* 5(4):110-120. doi: 10.11648/j.rd.20240504.13.
- Martínez, Francisco, Francisco Charte, Francisco A. Frías, dan Antonio J. Rivera. 2018. "Hybridizing Time Series Forecasting Models: A Review." *Progress in Artificial Intelligence* 7(3):161-174. doi: 10.1007/s40009-018-0275-5.
- Menculini, Lorenzo, Andrea Marini, Massimiliano Proietti, Alberto Garinei, Alessio Bozza, Cecilia Moretti, dan Marcello Marconi. 2021. "Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices." *Forecasting* 3(3):644-662. doi: 10.3390/forecast3030040.
- Papastefanopoulos, Vasilis, Pantelis Linardatos, dan Sotiris Kotsiantis. 2020. "COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population." *Applied Sciences* 10(11):3880. doi: 10.3390/app10113880.
- Pavitha, N. dan Shounak Sugave. 2023. "Optimizing Machine Learning Models: An Adaptive Hyperparameter Tuning Approach." *International Journal of Intelligent Systems and Applications in Engineering* 11(4):344-354. <https://ijisae.org/index.php/IJISAE/article/view/3532>.
- Prahutama, Alan, Tiani Wahyu Utami, and Hasbi Yasin. 2014. "Prediksi Harga Saham Menggunakan Support Vector Regression Dengan Algoritma Grid Search." *Media Statistika* 7(1):29-35. doi: <https://doi.org/10.14710/medstat.7.1.29-35>.
- Republik Indonesia. 2025. Peraturan Pemerintah Nomor 6 Tahun 2025 tentang Perubahan atas Peraturan Pemerintah Nomor 37 Tahun 2021 tentang Penyelenggaraan Program Jaminan Kehilangan Pekerjaan. Jakarta: Sekretariat Negara.

- 
- Samal, K. Krishna Rani, Korra Sathya Babu, Santosh Kumar Das, dan Abhirup Acharya. 2019. "Time Series based Air Pollution Forecasting using SARIMA and Prophet Model." *Proceedings of the 2019 International Conference on Information Technology and Computer Communications (ITCC '19)*: 80-85. Singapore: ACM. doi: 10.1145/3355402.3355417.
- Schuster, M., dan K. K. Paliwal. 1997. "Bidirectional recurrent neural networks." *IEEE Transactions on Signal Processing* 45(11):2673–2681.
- Shakeel, Asim, Daotong Chong, dan Jinshi Wang. 2023. "Load Forecasting of District Heating System Based on Improved FB-Prophet Model." *Energy* 278:127637. doi: 10.1016/j.energy.2023.127637.
- Shiri, Farhad Mortezapour, Thinagaran Perumal, Norwati Mustapha, dan Raihani Mohamed. 2023. "A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU." *arXiv*. doi: 10.48550/arXiv.2305.17646.
- Siami-Namini, Sima, Neda Tavakoli, dan Akbar Siami Namin. 2019. "The Performance of LSTM and BiLSTM in Forecasting Time Series." *2019 IEEE International Conference on Big Data (Big Data)*: 4385-4392. doi: 10.1109/BigData47094.2019.9005997.
- Taylor, Sean J., dan Benjamin Letham. 2018. "Forecasting at Scale." *The American Statistician* 72(1):37–45. doi: 10.1080/00031305.2017.1380080.
- Yan, Weizheng, Han Zhang, Jing Sui, dan Dinggang Shen. 2018. "Deep Chronnectome Learning via Full Bidirectional Long Short-Term Memory Networks for MCI Diagnosis." *Medical Image Computing and Computer Assisted Intervention* 249–257. doi: 10.1007/978-3-030-00931-1\_29.
- Yang, Li, dan Abdallah Shami. 2020. "On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice." *Neurocomputing* 415:295–316. doi: 10.1016/j.neucom.2020.07.061.
- Yenido˘ gan, I., C,ayir, A., et al. (2018). Bitcoin forecasting using ARIMA and Prophet. In 2018 3rd International Conference on Computer Science and Engineering (UBMK), 621–624.