

Perbandingan Hasil Peramalan XGBoost Tanpa Dan Dengan Optimisasi *Hyperparameter* Menggunakan *Whale Optimization Algorithm* Berbasis *Recursive Feature Elimination with Cross-Validation* (Studi Kasus: Data Curah Hujan Dasarian Kabupaten Pati)

Adeliya Fernanda^{1*}, Herlina Napitupulu², Nurul Gusriani³

^{1,2,3}Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Padjadjaran

Email: ^{1*}adeliya22001@mail.unpad.ac.id, ²herlina@unpad.ac.id, ³nurul.gusriani@unpad.ac.id

(* : coressponding author)

Abstrak – Kabupaten Pati merupakan salah satu sentra produksi garam yang proses produksinya masih bergantung pada kondisi curah hujan. Fenomena kemarau basah menyebabkan ketidakpastian waktu produksi sehingga diperlukan peramalan curah hujan yang akurat. Penelitian ini bertujuan memperoleh fitur yang paling berpengaruh, membandingkan performa model XGBoost tanpa dan dengan optimisasi *hyperparameter* menggunakan *Whale Optimization Algorithm* (WOA), serta menghasilkan peramalan curah hujan dasarian di Kabupaten Pati untuk enam dasarian ke depan. Penelitian diawali dengan *feature engineering* pada data historis curah hujan dasarian, dilanjutkan seleksi fitur menggunakan *Recursive Feature Elimination with Cross-Validation* (RFECV). Fitur terpilih digunakan untuk membangun model XGBoost tanpa optimisasi dan XGBoost-WOA. Performa model dievaluasi menggunakan *Root Mean Square Error* (RMSE). Hasil penelitian menunjukkan bahwa subset fitur optimal terdiri atas 12 fitur, yaitu *lag 1*, *lag 2*, *lag 3*, *lag 5*, *lag 6*, *rolling mean 3*, *rolling mean 6*, *rolling standard deviation 6*, *rolling mean 9*, *rolling mean 18*, *rolling standard deviation 18*, serta *dasarian sin*. Model XGBoost-WOA menghasilkan nilai RMSE sebesar 44,37, lebih baik dibandingkan XGBoost tanpa optimisasi sebesar 50,12. Peramalan enam dasarian ke depan berturut-turut sebesar 12,65; 32,77; 33,23; 42,66; 48,25; dan 48,25 mm/dasarian, yang menunjukkan kondisi musim kemarau masih berpotensi mendukung produksi garam meskipun terjadi peningkatan curah hujan pada akhir periode. Pendekatan XGBoost-WOA berpotensi menjadi alternatif metode peramalan curah hujan dasarian untuk mendukung perencanaan produksi garam di Kabupaten Pati.

Kata Kunci: XGBoost, *Whale Optimization Algorithm*, *Recursive Feature Elimination with Cross-Validation*, Peramalan Curah Hujan Dasarian, Kabupaten Pati

Abstract – Pati Regency is one of Indonesia's major salt-producing regions, where production remains highly dependent on rainfall conditions. The wet dry-season phenomenon has increased uncertainty in production schedules, highlighting the need for accurate rainfall forecasting. This study aims to identify the most influential features, compare the performance of XGBoost models with and without hyperparameter optimization using the *Whale Optimization Algorithm* (WOA), and forecast ten-day rainfall in Pati Regency for the next six periods. The research began with feature engineering on historical ten-day rainfall data, followed by feature selection using *Recursive Feature Elimination with Cross-Validation* (RFECV). The selected features were used to develop both the baseline XGBoost model and the WOA-optimized XGBoost model. Model performance was evaluated using the *Root Mean Square Error* (RMSE). The optimal feature subset consisted of 12 features: *lag 1*, *lag 2*, *lag 3*, *lag 5*, *lag 6*, *rolling mean 3*, *rolling mean 6*, *rolling standard deviation 6*, *rolling mean 9*, *rolling mean 18*, *rolling standard deviation 18*, and the *dasarian sine* feature. The optimized XGBoost-WOA model achieved a lower RMSE (44.37) than the baseline XGBoost model (50.12). Forecasted rainfall for the next six ten-day periods was 12.65, 32.77, 33.23, 42.66, 48.25, and 48.25 mm per ten-day period, indicating that dry-season conditions remain favorable for salt production despite increasing rainfall toward the end of the forecast horizon. Therefore, XGBoost-WOA provides a promising alternative for ten-day rainfall forecasting to support salt production planning in Pati Regency.

Keywords: XGBoost, *Whale Optimization Algorithm*, *Recursive Feature Elimination with Cross-Validation*, ten-day rainfall forecasting, Pati Regency.

1. PENDAHULUAN

Kabupaten Pati merupakan salah satu sentra produksi garam di Indonesia yang proses produksinya masih bergantung pada kondisi curah hujan (Damayanti et al., 2025). Pada tahun 2025, produksi garam di Kabupaten Pati mengalami penurunan sekitar 21,94% akibat fenomena kemarau

basah, yaitu kondisi ketika curah hujan tetap berada di atas normal meskipun telah memasuki musim kemarau (Hanafi, 2025; Kementerian Kelautan dan Perikanan, 2025; Zahro, 2025). Kondisi tersebut menyebabkan ketidakpastian waktu produksi sehingga diperlukan peramalan curah hujan yang akurat sebagai dasar perencanaan produksi garam.

Berbagai metode telah digunakan dalam peramalan curah hujan, contohnya ARIMA (Mislan & Dani, 2025). Namun, metode statistik tersebut memiliki keterbatasan karena memerlukan pemenuhan asumsi tertentu (Kong et al., 2025). Oleh karena itu, metode *machine learning* seperti XGBoost banyak digunakan karena lebih fleksibel. Penelitian sebelumnya menunjukkan bahwa XGBoost memberikan performa peramalan curah hujan yang lebih baik dibandingkan beberapa algoritma *machine learning* lainnya (Kumar et al., 2023).

Performa XGBoost sangat dipengaruhi oleh pemilihan *hyperparameter* (Bahmani et al., 2021). Metode optimisasi konvensional, seperti *Grid Search* dan *Random Search* memiliki keterbatasan. *Grid Search* tidak efisien pada ruang pencarian berdimensi tinggi, sedangkan *Random Search* tidak memanfaatkan hasil iterasi sebelumnya sehingga pencarian solusi optimal kurang efisien (Hutter et al., 2019). Oleh karena itu, berbagai penelitian mulai memanfaatkan algoritma optimisasi berbasis *metaheuristic* karena mampu menangani permasalahan kompleks berskala besar dengan beban komputasi yang relatif rendah serta menghindari terjebak pada optimum lokal (S. B. Joseph et al., 2022; Mirjalili & Lewis, 2016). Salah satu algoritma *metaheuristic* yang banyak digunakan adalah *Whale Optimization Algorithm* (WOA). Qiu et al. (2021) menunjukkan bahwa WOA mampu meningkatkan performa XGBoost dibandingkan beberapa metode optimisasi dan algoritma *machine learning* lainnya.

Selain optimisasi *hyperparameter*, data curah hujan sebagai data *time series* memerlukan *feature engineering* untuk menangkap pola temporal (Balasubramaniam et al., 2025). Penambahan fitur tersebut dapat meningkatkan kompleksitas model sehingga diperlukan pemilihan fitur yang mampu mempertahankan fitur-fitur yang paling relevan. Metode *Recursive Feature Elimination with Cross Validation* (RFECV) mampu meningkatkan performa model melalui pemilihan *subset* fitur yang optimal (Kumari et al., 2025).

Berdasarkan uraian tersebut, penelitian ini bertujuan meramalkan curah hujan dasarian di Kabupaten Pati menggunakan algoritma XGBoost. Pemodelan diawali dengan *feature engineering* dan pemilihan fitur menggunakan *Recursive Feature Elimination with Cross Validation* (RFECV), kemudian fitur terpilih digunakan untuk membangun dua model, yaitu XGBoost tanpa optimisasi dan XGBoost dengan *hyperparameter* yang dioptimisasi menggunakan *Whale Optimization Algorithm* (WOA). Performa kedua model dibandingkan menggunakan *Root Mean Square Error* (RMSE), kemudian model terbaik digunakan untuk meramalkan curah hujan enam dasarian ke depan.

2. METODE

2.1 Curah Hujan

Curah hujan didefinisikan sebagai jumlah air hujan yang mencapai suatu wilayah dalam rentang waktu tertentu dan biasanya diukur dalam satuan milimeter (mm). Nilai curah hujan sebesar 1 mm menunjukkan bahwa setiap satu meter persegi luas permukaan menerima air hujan dengan volume yang setara dengan satu liter (BMKG, 2025). Salah satu periode yang umum digunakan dalam analisis curah hujan adalah dasarian, yaitu periode 10 harian yang terdiri atas dasarian I (tanggal 1–10), dasarian II (tanggal 11–20), dan dasarian III (tanggal 21 hingga akhir bulan) (Zahro, 2025). Berdasarkan BMKG, curah hujan dasarian dikelompokkan menjadi empat kategori, yaitu rendah (0–50 mm/dasarian), menengah (50–150 mm/dasarian), tinggi (150–300 mm/dasarian), dan sangat tinggi (>300 mm/dasarian). Curah hujan dasarian juga digunakan sebagai dasar penentuan awal musim, yaitu musim kemarau apabila curah hujan kurang dari 50 mm/dasarian selama tiga dasarian berturut-turut, dan musim hujan apabila curah hujan lebih dari atau sama dengan 50 mm/dasarian selama tiga dasarian berturut-turut (Saputri, 2025).

2.2. Feature Engineering

Feature engineering merupakan proses mengubah data mentah menjadi fitur yang sesuai untuk model *machine learning*. Pada data *time series*, teknik ini digunakan untuk membentuk fitur yang mampu menangkap pola temporal dalam data (M. Joseph, 2022). Teknik yang umum digunakan meliputi *lag features*, *rolling window aggregations*, *calendar features*, dan *cyclical features*. *Lag features* menggunakan nilai periode sebelumnya yang dirumuskan sebagai:

$$Lag_{o,i} = y_{i-o}, \quad (1)$$

dengan y_i merupakan nilai variabel pada pengamatan ke- i , sedangkan o merupakan orde *lag*. *Rolling window aggregations* memanfaatkan beberapa observasi historis melalui perhitungan agregasi, seperti *mean* dan *standard deviation*, yang dirumuskan sebagai:

$$Rollmean_{\omega,i} = \frac{1}{\omega} \sum_{u=1}^{\omega} y_{i-u}, \quad (2)$$

$$Rollstd_{\omega,i} = \sqrt{\frac{1}{\omega} \sum_{u=1}^{\omega} (y_{i-u} - Rollmean_{\omega,i})^2}, \quad (3)$$

dengan ω merupakan ukuran jendela waktu. *Calendar features* digunakan untuk mengekstraksi informasi waktu guna menangkap pola musiman.

Untuk mengatasi diskontinuitas pada fitur waktu, *calendar features* dapat ditransformasikan menjadi *cyclical features* menggunakan fungsi *sinus* dan *cosinus* (Khazem & Kanso, 2025), yaitu:

$$Cal_sin_i = \sin \sin \left(\frac{2\pi \times Cal_i}{TimeCycle} \right) \quad \text{dan} \quad Cal_cos_i = \cos \cos \left(\frac{2\pi \times Cal_i}{TimeCycle} \right), \quad (4)$$

dengan Cal_i merupakan nilai *calendar features* pada observasi ke- i , sedangkan $TimeCycle$ merupakan panjang periode satu siklus.

2.3. eXtreme Gradient Boosting.

XGBoost (*eXtreme Gradient Boosting*) merupakan pengembangan dari *Gradient Boosting* yang diperkenalkan oleh Chen & Guestrin (2016) sebagai algoritma *tree boosting* yang skalabel dan efisien. XGBoost membentuk model secara aditif dengan menambahkan *regression tree* baru pada setiap iterasi untuk memperbaiki kesalahan prediksi model sebelumnya. Model pada iterasi ke- t dinyatakan sebagai:

$$F^{(t)}(x) = F^{(t-1)}(x) + f^{(t)}(x), \quad (5)$$

dengan $F^{(t)}(x)$ merupakan model prediksi pada iterasi ke- t , sedangkan $f^{(t)}(x)$ merupakan *regression tree* yang ditambahkan pada iterasi ke- t .

Proses pembentukan model dilakukan dengan meminimalkan fungsi objektif, yaitu:

$$L^{(t)} = \sum_{i=1}^I l(y_i, F^{(t)}(x_i)) + \Omega(f^{(t)}), \quad (6)$$

dengan $l(y_i, F^{(t)}(x_i))$ merupakan nilai *loss* antara nilai aktual y_i dan nilai prediksi $F^{(t)}(x_i)$, sedangkan $\Omega(f^{(t)})$ merupakan fungsi regularisasi untuk *regression tree* yang ditambahkan pada iterasi ke- t . Untuk mempermudah proses optimisasi, fungsi objektif dapat diaproksimasi

menggunakan *second-order Taylor expansion* sehingga diperoleh gradien orde pertama $g_i^{(t)}$ dan gradien orde kedua $h_i^{(t)}$ sebagai dasar pembentukan *regression tree*.

Berdasarkan pendekatan tersebut, bobot *leaf node* optimal diperoleh sebagai:

$$w_v^{*(t)} = - \frac{\sum_{i \in I_v^{(t)}} g_i^{(t)}}{\sum_{i \in I_v^{(t)}} h_i^{(t)} + \lambda}, \quad (7)$$

dengan $I_v^{(t)}$ merupakan himpunan pengamatan pada *leaf node* ke- v iterasi ke- t , sedangkan λ merupakan parameter regularisasi. Kualitas setiap pemisahan (*split*) dievaluasi menggunakan nilai *gain*, yaitu:

$$L_s^{(t)} = \frac{1}{2} \left(\frac{\left(\sum_{i \in I_l^{(t)}} g_i^{(t)} \right)^2}{\sum_{i \in I_l^{(t)}} h_i^{(t)} + \lambda} + \frac{\left(\sum_{i \in I_r^{(t)}} g_i^{(t)} \right)^2}{\sum_{i \in I_r^{(t)}} h_i^{(t)} + \lambda} - \frac{\left(\sum_{i \in I^{(t)}} g_i^{(t)} \right)^2}{\sum_{i \in I^{(t)}} h_i^{(t)} + \lambda} \right) - \gamma. \quad (8)$$

Pemisahan terbaik dipilih berdasarkan nilai *gain* terbesar sehingga menghasilkan struktur *regression tree* yang optimal (Chen & Guestrin, 2016).

2.4 Whale Optimization Algorithm

WOA (*Whale Optimization Algorithm*) diperkenalkan oleh Mirjalili & Lewis (2016) sebagai algoritma optimisasi metaheuristik yang terinspirasi dari perilaku berburu paus bungkuk (*Megaptera novaeangliae*). WOA memodelkan perilaku paus dalam tiga mekanisme utama, yaitu:

1. *Encircling Prey*.

Mekanisme *encircling prey* menggambarkan perilaku paus dalam mendeteksi dan mengelilingi mangsa. Dalam WOA, paus direpresentasikan sebagai agen pencarian (*search agent*), sedangkan posisi terbaik yang diperoleh pada suatu iterasi dianggap sebagai perkiraan posisi mangsa. Agen pencarian lainnya akan memperbarui posisi menuju solusi terbaik tersebut. Misalkan $\vec{U}_p^{(\tau)}$ menyatakan vektor posisi agen pencarian ke- p pada iterasi ke- τ dengan $p = 1, 2, \dots, P$. Jika ruang pencarian memiliki M dimensi, maka $\vec{U}_p^{(\tau)}$ dapat dituliskan sebagai:

$$\vec{U}_p^{(\tau)} = (U_{1,p}^{(\tau)} \ U_{1,p}^{(\tau)} : U_{M,p}^{(\tau)}). \quad (9)$$

Jarak agen pencarian terhadap solusi terbaik $\vec{U}^{*(\tau)}$ dirumuskan sebagai:

$$\vec{D}_p^{(\tau)} = \left| \vec{C}_p^{(\tau)} \cdot \vec{U}^{*(\tau)} - \vec{U}_p^{(\tau)} \right|. \quad (10)$$

Selanjutnya, posisi agen diperbarui menggunakan:

$$\vec{U}_p^{(\tau+1)} = \vec{U}^{*(\tau)} - \vec{A}_p^{(\tau)} \cdot \vec{D}_p^{(\tau)}, \quad (11)$$

dengan vektor koefisien $\vec{A}_p^{(\tau)}$ dan $\vec{C}_p^{(\tau)}$ diperoleh dari:

$$\vec{A}_p^{(\tau)} = 2\vec{a}^{(\tau)} \cdot \vec{r}_{1,p}^{(\tau)} - \vec{a}^{(\tau)} \text{ dan } \vec{C}_p^{(\tau)} = 2 \cdot \vec{r}_{2,p}^{(\tau)}. \quad (12)$$

Vektor $\vec{a}^{(\tau)}$ mengalami penurunan linier dari 2 menuju 0 selama proses iterasi, yaitu:

$$\vec{a}^{(\tau)} = (a_1^{(\tau)} \ a_2^{(\tau)} \ : \ a_M^{(\tau)}), \quad a_1^{(\tau)} = a_2^{(\tau)} = \dots = a_M^{(\tau)} = 2 - \tau \left(\frac{2}{\tau_{max}} \right), \quad (13)$$

sedangkan $\vec{r}_{1,p}^{(\tau)}$ dan $\vec{r}_{2,p}^{(\tau)}$ merupakan vektor bilangan acak pada interval $[0,1]$.

2. Bubble-net Attacking Method.

Mekanisme ini terdiri atas dua pendekatan, yaitu *shrinking encircling mechanism* dan *spiral updating position*. *Shrinking encircling mechanism* dilakukan dengan menurunkan nilai elemen $\vec{a}^{(\tau)}$ secara bertahap dari 2 menjadi 0 sehingga ruang pencarian semakin menyempit. Sementara itu, *spiral updating position* memodelkan pergerakan spiral paus menuju mangsa, yaitu:

$$\vec{U}_p^{(\tau+1)} = \vec{D}_p^{sprl(\tau)} \cdot \exp(bl_p^{(\tau)}) \cdot \cos \cos(2\pi l_p^{(\tau)}) + \vec{U}^{(\tau)}, \quad (14)$$

dengan:

$$\vec{D}_p^{sprl(\tau)} = |\vec{U}^{(\tau)} - \vec{U}_p^{(\tau)}|. \quad (15)$$

Parameter b adalah konstanta yang menentukan bentuk spiral umumnya bernilai 1 (Sun et al., 2022) dan $l_p^{(\tau)}$ merupakan bilangan acak pada interval $[-1,1]$. Kedua mekanisme tersebut dipilih secara probabilistik dengan peluang sebesar 50%, sehingga posisi agen diperbarui menggunakan:

$$\begin{aligned} \vec{U}_p^{(\tau+1)} &= \{\vec{U}^{(\tau)} - \vec{A}_p^{(\tau)} \cdot \vec{D}_p^{(\tau)}, \quad z_p^{(\tau)} \\ &< 0,5, \vec{D}_p^{sprl(\tau)} \cdot \exp(bl_p^{(\tau)}) \cdot \cos \cos(2\pi l_p^{(\tau)}) + \vec{U}^{(\tau)}, \quad z_p^{(\tau)} \geq 0,5, \end{aligned} \quad (16)$$

dengan $z_p^{(\tau)}$ bilangan acak pada interval $[0,1]$.

3. Search for Prey.

Mekanisme ini digunakan untuk mengeksplorasi ruang pencarian yang lebih luas. Berbeda dengan mekanisme sebelumnya yang menggunakan solusi terbaik sebagai acuan, mekanisme ini menggunakan posisi agen pencarian yang dipilih secara acak dari populasi. Mekanisme ini terjadi ketika nilai $|\vec{A}^{(\tau)}| \geq 1$. Misalkan $\vec{U}_{rndm}^{(\tau)}$ merupakan posisi agen yang dipilih secara acak, maka jarak terhadap agen pencarian ke- p dirumuskan sebagai:

$$\vec{D}_p^{rndm(\tau)} = |\vec{C}_p^{(\tau)} \cdot \vec{U}_{rndm}^{(\tau)} - \vec{U}_p^{(\tau)}| \quad (17)$$

dan pembaruan posisi dilakukan dengan:

$$\vec{U}_p^{(\tau+1)} = \vec{U}_{rndm}^{(\tau)} - \vec{A}_p^{(\tau)} \cdot \vec{D}_p^{rndm(\tau)}. \quad (18)$$

2.5. Time Series Split

Cross-validation digunakan untuk memperkirakan kesalahan generalisasi model (Hastie et al., 2009). Pada data *time series*, *Time Series Split* digunakan untuk membagi data secara berurutan berdasarkan waktu sehingga data masa lalu digunakan untuk memprediksi data berikutnya (Pedregosa et al., 2011). Kesalahan generalisasi model diperoleh berdasarkan rata-rata hasil evaluasi dari setiap pembagian data.

2.6. Recursive Feature Elimination with Cross-Validation

Pemilihan fitur merupakan tahap penting dalam model *machine learning* karena dapat mengurangi dimensionalitas data dan meningkatkan performa model (Koller & Sahami, 1996). RFECV (*Recursive Feature Elimination with Cross-Validation*) merupakan pengembangan dari RFE (*Recursive Feature Elimination*) yang bekerja dengan melatih model, menghitung tingkat kepentingan fitur, dan mengeliminasi fitur dengan kepentingan terendah secara iteratif (Guyon et al., 2002). Pada RFECV, jumlah fitur optimal ditentukan melalui evaluasi beberapa subset fitur pada setiap *fold cross-validation*. Setelah jumlah fitur optimal diperoleh, proses RFE dilakukan kembali menggunakan seluruh data *training* untuk menentukan *subset* fitur akhir (Guyon et al., 2002).

2.7. Root Mean Square Error

Metrik evaluasi digunakan untuk menilai performa model dengan membandingkan nilai peramalan dan nilai aktual. Salah satu metrik yang umum digunakan adalah *Root Mean Square Error* (RMSE) (Hyndman & Koehler, 2006), yang dirumuskan sebagai:

$$RMSE = \sqrt{\frac{1}{I} \sum_{i=1}^I (y_i - \hat{y}_i)^2}, \quad i = 1, 2, \dots, I, \quad (19)$$

dengan y_i merupakan nilai aktual ke- i , $\hat{y}_i$ merupakan nilai peramalan ke- i , dan I adalah banyaknya pengamatan. Nilai RMSE yang lebih kecil menunjukkan kesalahan peramalan yang lebih rendah. RMSE banyak digunakan dalam evaluasi model peramalan, termasuk bidang iklim dan lingkungan (Willmott & Matsuura, 2005).

2.8. Data Penelitian

Data diperoleh dari *website NASA Prediction of Worldwide Energy Resources (POWER)*. Data diambil pada grid spasial $0,5^\circ \times 0,625^\circ$ dengan titik pusat koordinat $-6,7083^\circ$ LS dan $111,0417^\circ$ BT yang merepresentasikan wilayah Kabupaten Pati. Periode data yang digunakan adalah 21 Agustus 2016 hingga 10 Juli 2026. Data yang diperoleh dari NASA POWER berupa data curah hujan harian, kemudian ditransformasi menjadi data curah hujan dasarian sehingga diperoleh 356 observasi.

2.6. Recursive Feature Elimination with Cross-Validation

Pemilihan fitur merupakan tahap penting dalam model *machine learning* karena dapat mengurangi dimensionalitas data dan meningkatkan performa model (Koller & Sahami, 1996). RFECV (*Recursive Feature Elimination with Cross-Validation*) merupakan pengembangan dari RFE (*Recursive Feature Elimination*) yang bekerja dengan melatih model, menghitung tingkat kepentingan fitur, dan mengeliminasi fitur dengan kepentingan terendah secara iteratif (Guyon et al., 2002). Pada RFECV, jumlah fitur optimal ditentukan melalui evaluasi beberapa subset fitur pada setiap *fold cross-validation*. Setelah jumlah fitur optimal diperoleh, proses RFE dilakukan kembali menggunakan seluruh data *training* untuk menentukan *subset* fitur akhir (Guyon et al., 2002).

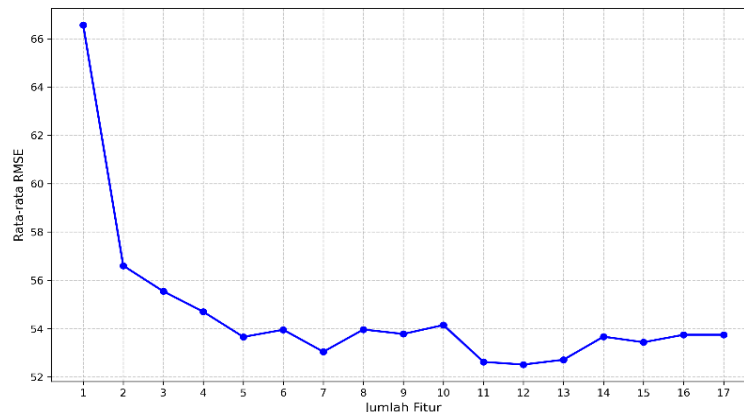
3. HASIL DAN PEMBAHASAN

3.1 Data Preprocessing

Data *preprocessing* dilakukan untuk mempersiapkan data sebelum proses peramalan. Tahapan ini meliputi *feature engineering*, penghapusan *missing value*, dan pembagian data. *Feature engineering* menghasilkan 17 fitur yang terdiri dari *lag features* sebanyak 6 fitur (*lag* 1 hingga *lag* 6), *rolling window aggregations* sebanyak 8 fitur yang meliputi *rolling mean* dan *rolling standard deviation* dengan ukuran jendela 3, 6, 9, dan 18 dasarian, serta *calendar features* berupa dasarian dan *cyclical features* berupa transformasi *sinus* dan *cosinus* dari fitur dasarian. Fitur yang terbentuk kemudian menyebabkan adanya *missing value* pada observasi awal sehingga dilakukan penghapusan data dan menghasilkan 338 observasi. Selanjutnya, data dibagi menggunakan rasio 80% sebagai data *training* dan 20% sebagai data *testing*.

3.2 Pemilihan Fitur dengan RFECV

Pemilihan fitur dilakukan menggunakan RFECV yang diimplementasikan dengan bahasa pemrograman Python menggunakan *library* scikit-learn. RFECV diterapkan pada data *training* dengan parameter *estimator* menggunakan XGBoost sebagai model dasar, *step* sebesar 1 yang berarti satu fitur dengan *feature importance* terendah dieliminasi pada setiap iterasi, *min_features_to_select* sebesar 1 yang menunjukkan proses eliminasi dihentikan ketika tersisa satu fitur, *cv* menggunakan *Time Series Split 5 fold* untuk mempertahankan urutan waktu pada data, serta *scoring* menggunakan RMSE untuk mengevaluasi kinerja model pada setiap iterasi.



Gambar 1. Grafik rata-rata RMSE terhadap jumlah fitur pada proses RFECV

Gambar 1 menunjukkan grafik rata-rata RMSE hasil *cross-validation* 5-fold terhadap jumlah fitur yang digunakan selama proses RFECV. Titik rata-rata RMSE terendah dicapai pada 12 fitur, sehingga *subset* fitur akhir ditentukan pada jumlah tersebut dan digunakan untuk melatih model pada seluruh data *training*. Kedua belas fitur tersebut adalah *lag* 1, *lag* 2, *lag* 3, *lag* 5, *lag* 6, *rolling mean* 3, *rolling mean* 6, *rolling standard deviation* 6, *rolling mean* 9, *rolling mean* 18, *rolling standard deviation* 18, dan *dasar* *sin*.

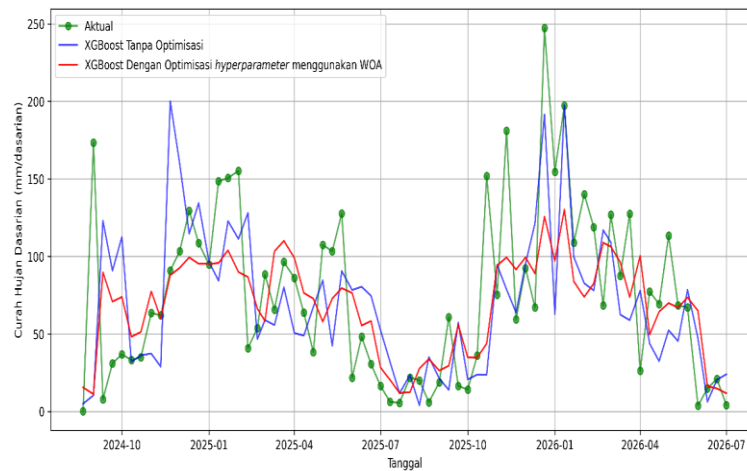
3.3 Peramalan XGBoost tanpa Optimisasi *Hyperparameter*

Kedua belas fitur optimal yang diperoleh pada tahap sebelumnya digunakan untuk membangun model XGBoost tanpa optimisasi *hyperparameter* pada data *training*. Model tersebut diimplementasikan menggunakan *library* XGBoost dengan *hyperparameter default* yang telah ditetapkan pada *library* tersebut. Model XGBoost yang terbentuk terdiri atas 100 *regression tree*. Selanjutnya, model tersebut digunakan untuk menghasilkan nilai peramalan pada data *testing*.

3.4 Peramalan XGBoost dengan Optimisasi *Hyperparameter* Menggunakan WOA

Peramalan XGBoost dengan optimisasi *hyperparameter* menggunakan WOA dilakukan dengan mengoptimalkan empat *hyperparameter*, yaitu *n_estimators*, *max_depth*, *gamma*, dan *learning_rate*. Optimisasi tersebut dilakukan dengan parameter WOA berupa jumlah paus ($P = 50$), iterasi maksimum ($\tau_{max} = 150$), dimensi pencarian ($M = 4$), dan konstanta spiral ($b = 1$). Setiap posisi paus merepresentasikan kombinasi *hyperparameter* yang dievaluasi berdasarkan nilai *fitness* berupa rata-rata RMSE dari 5 *fold Time Series Split*. Berdasarkan proses iterasi WOA, solusi optimal diperoleh pada iterasi ke-65 karena tidak terdapat perbaikan nilai *fitness* selama 25 iterasi berturut-turut. Kombinasi *hyperparameter* optimal yang diperoleh yaitu *n_estimators*= 80, *max_depth*= 1, *gamma*=0,11, dan *learning_rate*= 0,09. Kombinasi tersebut kemudian digunakan untuk membangun model XGBoost pada data *training* yang terdiri atas 80 *regression tree* dan selanjutnya digunakan untuk menghasilkan nilai peramalan pada data *testing*.

3.5 Evaluasi dan Perbandingan Performa Model.



Gambar 2. Grafik hasil peramalan model XGBoost tanpa dan dengan optimisasi hyperparameter menggunakan WOA

Berdasarkan Gambar 2, kedua model XGBoost mampu mengikuti pola curah hujan aktual, tetapi model XGBoost dengan optimisasi *hyperparameter* menggunakan WOA menghasilkan peramalan yang lebih stabil. Meskipun masih mengalami keterbatasan dalam menangkap curah hujan ekstrem, hasil evaluasi pada TABEL 1. menunjukkan bahwa model dengan optimisasi WOA memiliki nilai RMSE lebih rendah, yaitu 44,37 dibandingkan model tanpa optimisasi sebesar 50,12. Oleh karena itu, model XGBoost dengan optimisasi WOA dipilih sebagai model terbaik.

Tabel 1. Hasil evaluasi performa model

Model	RMSE
XGBoost tanpa optimisasi <i>hyperparameter</i>	50,12
XGBoost dengan optimisasi <i>hyperparameter</i> menggunakan WOA	44,37

3.6 Hasil Peramalan Curah Hujan 6 Dasarian ke Depan.

Tabel 2. Hasil peramalan curah hujan 6 dasarian ke depan

Tanggal	Curah Hujan Dasarian (mm/dasarian)
11/07/2026	12,65
21/07/2026	32,77
01/08/2026	33,23
11/08/2026	42,66
21/08/2026	48,25
01/09/2026	48,25

Berdasarkan hasil evaluasi, model XGBoost dengan optimisasi *hyperparameter* menggunakan WOA digunakan untuk melakukan peramalan curah hujan 6 dasarian ke depan. Hasil peramalan pada Tabel 2. menunjukkan nilai curah hujan berada pada rentang 12,65–48,25 mm/dasarian dan masih berada di bawah ambang batas 50 mm/dasarian sehingga Kabupaten Pati diramalkan masih mengalami musim kemarau. Meskipun demikian, peningkatan curah hujan pada akhir periode perlu diperhatikan sebagai informasi dalam penyesuaian jadwal produksi garam.

3.7 Pembahasan

Hasil penelitian menunjukkan bahwa penerapan pemilihan fitur menggunakan RFECV dan optimisasi *hyperparameter* menggunakan WOA mampu meningkatkan performa model XGBoost dalam peramalan curah hujan dasarian. RFECV memilih 12 fitur terbaik yang didominasi oleh *lag features* dan *rolling window aggregations*. Hal ini menunjukkan bahwa informasi curah hujan pada beberapa dasarian sebelumnya, serta pola rata-rata dan variasinya merupakan informasi yang paling berpengaruh dalam membentuk model peramalan.

Optimisasi *hyperparameter* menggunakan WOA menghasilkan kombinasi $n_estimators=80$, $max_depth=1$, $gamma=0,11$, dan $learning_rate=0,09$. Dibandingkan dengan penggunaan *hyperparameter default*, kombinasi tersebut menghasilkan model yang lebih sederhana, tetapi mampu memberikan hasil peramalan yang lebih baik pada data *testing*. Nilai max_depth yang rendah membatasi kompleksitas pohon yang dibentuk sehingga dapat mengurangi kecenderungan model mengalami *overfitting*.

Meskipun demikian, berdasarkan grafik hasil peramalan, kedua model masih mengalami kesulitan dalam mengikuti lonjakan curah hujan yang tinggi. Kondisi ini menunjukkan bahwa data curah hujan memiliki variabilitas yang tinggi sehingga kejadian hujan ekstrem masih menjadi tantangan dalam proses peramalan. Namun demikian, optimisasi *hyperparameter* menggunakan WOA tetap mampu menghasilkan peramalan yang lebih mendekati data aktual dibandingkan model tanpa optimisasi.

4. KESIMPULAN

Berdasarkan hasil penelitian, RFECV menghasilkan 12 fitur optimal, yaitu *lag 1*, *lag 2*, *lag 3*, *lag 5*, *lag 6*, *rolling mean 3*, *rolling mean 6*, *rolling standard deviation 6*, *rolling mean 9*, *rolling mean 18*, *rolling standard deviation 18*, dan *dasarian sin* sebagai fitur yang paling berpengaruh dalam peramalan curah hujan dasarian di Kabupaten Pati. Hasil evaluasi menunjukkan bahwa model XGBoost dengan optimisasi *hyperparameter* menggunakan WOA memberikan performa yang lebih baik dibandingkan XGBoost tanpa optimisasi, dengan nilai RMSE masing-masing sebesar 44,37 dan 50,12. Model terbaik tersebut kemudian digunakan untuk meramalkan curah hujan enam dasarian ke depan dengan hasil berturut-turut sebesar 12,65, 32,77, 33,23, 42,66, 48,25, dan 48,25 mm/dasarian. Seluruh hasil peramalan berada di bawah ambang batas 50 mm/dasarian, sehingga Kabupaten Pati diperkirakan masih berada pada kondisi musim kemarau yang berpotensi mendukung produksi garam, meskipun peningkatan curah hujan pada akhir periode perlu diperhatikan dalam penentuan waktu produksi dan panen garam.

REFERENCES

- Bahmani, M., Shawi, R. El, Potikyan, N., & Sakr, S. (2021). *To tune or not to tune? An Approach for Recommending Important Hyperparameters*. <https://doi.org/10.1145/1122445.1122456>
- Balasubramaniam, T., Mohotti, W. A., Sabir, K., & Nayak, R. (2025). Feature engineering on climate data with machine learning to understand time-lagging effects in pasture yield prediction. *Ecological Informatics*, 86, 103011. <https://doi.org/10.1016/J.ECOINF.2025.103011>
- BMKG. (2025). *Curah Hujan*. Badan Meteorologi, Klimatologi, dan Geofisika. <https://gawbariri.bmkg.go.id/index.php/karya-tulis-dan-artikel/gawsarium/262-curah-hujan>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 13-17-August-2016*, 785–794. <https://doi.org/10.1145/2939672.2939785>;CSUBTYPE:STRING:CONFERENCE
- Damayanti, H. O., Ernawati, A., Widhayoko, Astuti, A. D., & Aini, S. Q. (2025). Tata Kelola Usaha Garam Rakyat di Kabupaten Pati. In *Penelitian. Sistem Informasi Riset dan Inovasi Daerah (SIRIDA) Kabupaten Pati*. <https://sirida.patikab.go.id/riset/bc16baaa-61df-4497-9e7e-015ec46363f9>
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1–3), 389–422. <https://doi.org/10.1023/A:1012487302797/METRICS>
- Hanafi, U. (2025). *Kemarau Basah, Produksi Garam di Pati Anjlok*. <https://berita.murianews.com/umar-hanafi/449702/kemarau-basah-produksi-garam-di-pati-anjlok>

- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer. <https://doi.org/10.1007/978-0-387-84858-7>
- Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). Automated Machine Learning. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), *Methods, Systems, Challenges*. Springer. <https://doi.org/10.1007/978-3-030-05318-5>
- Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688. <https://doi.org/10.1016/J.IJFORECAST.2006.03.001>
- Joseph, M. (2022). *Modern Time Series Forecasting with Python*. Packt Publishing. <https://www.oreilly.com/library/view/modern-time-series/9781803246802/>
- Joseph, S. B., Dada, E. G., Abidemi, A., Oyewola, D. O., & Khammas, B. M. (2022). Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems. *Heliyon*, 8(5), e09399. <https://doi.org/10.1016/J.HELİYON.2022.E09399>
- Kementerian Kelautan dan Perikanan. (2025). *Produksi Garam*. Pusat Data, Statistik, dan Informasi, Kementerian Kelautan dan Perikanan. https://portaldata.kkp.go.id/portals/data-statistik/prod_garam/summary
- Khazem, S., & Kalso, H. (2025). *Cyclical Temporal Encoding and Hybrid Deep Ensembles for Multistep Energy Forecasting*. <https://arxiv.org/pdf/2512.03656>
- Koller, D., & Sahami, M. (1996). Toward Optimal Feature Selection. *Proceedings of the Thirteenth International Conference on Machine Learning*, 284–292. <https://dl.acm.org/doi/10.5555/3091696.3091731>
- Kong, X., Chen, Z., Liu, W., Ning, K., Zhang, L., Muhammad Marier, S., Liu, Y., Chen, Y., & Xia, F. (2025). Deep learning for time series forecasting: a survey. *International Journal of Machine Learning and Cybernetics* 2025 16:7, 16(7), 5079–5112. <https://doi.org/10.1007/S13042-025-02560-W>
- Kumar, V., Kedam, N., Sharma, K. V., Khedher, K. M., & Alluqmani, A. E. (2023). A Comparison of Machine Learning Models for Predicting Rainfall in Urban Metropolitan Cities. *Sustainability* 2023, Vol. 15, Page 13724, 15(18), 13724. <https://doi.org/10.3390/SU151813724>
- Kumari, P., Harshith, N., & Ginige, A. (2025). Integrating recursive feature selection with automated machine learning framework for global wheat price prediction. *Journal of Agriculture and Food Research*, 22, 102113. <https://doi.org/10.1016/J.JAFR.2025.102113>
- Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51–67. <https://doi.org/10.1016/J.ADVENGSOFT.2016.01.008>
- Mislan, & Dani, A. T. R. (2025). Navigating Samarinda's climate: A comparative analysis of rainfall forecasting models. *MethodsX*, 14, 103080. <https://doi.org/10.1016/J.MEX.2024.103080>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. <https://jmlr.org/papers/v12/pedregosa11a.html>
- Qiu, Y., Zhou, J., Khandelwal, M., Yang, H., Yang, P., & Li, C. (2021). Performance evaluation of hybrid WOA-XGBoost, GWO-XGBoost and BO-XGBoost models to predict blast-induced ground vibration. *Engineering with Computers* 2021 38:5, 38(5), 4145–4162. <https://doi.org/10.1007/S00366-021-01393-9>
- Saputri, R. E. (2025). *Prediksi Musim Kemarau Tahun 2025 di Indonesia (Pemutakhiran Mei 2025)*. Badan Meteorologi, Klimatologi, dan Geofisika. <https://www.bmkg.go.id/iklim/prediksi-musim/prediksi-musim-kemarau-tahun-2025-di-indonesia-pemutakhiran-mei-2025-2>
- Sun, G., Shang, Y., Yuan, K., & Gao, H. (2022). An improved whale optimization algorithm based on nonlinear parameters and feedback mechanism. *International Journal of Computational Intelligence Systems*, 15, 38. <https://doi.org/10.1007/s44196-022-00092-7>
- Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1), 79–82. <https://doi.org/10.3354/CR030079>
- Zahro, A. A. (2025). *BMKG: Musim Kemarau Basah Diprediksi Hingga Oktober 2025, Waspada Bencana Hidrometeorologi*. <https://www.bmkg.go.id/berita/utama/bmkg-musim-kemarau-basah-diprediksi-hingga-oktober-2025-waspada-bencana-hidrometeorologi>