

Perbandingan Metode *Hyperparameter Tuning* Pada Model XGBoost Untuk Prediksi *Multioutput*

Lukman Handyanto¹, Willdan Aprizal Arifin^{*2}, Herman Syafri³

^{1,2,3}Universitas Pendidikan Indonesia, Jl. Dr. Setiabudi No.229, Bandung, Jawa Barat, Indonesia
e-mail: ¹lhandyanto@upi.edu, ^{*2}willdanarifin@upi.edu, ³herman.syafri@upi.edu

Abstrak

Prediksi parameter atmosfer-oseanografi yang akurat sangat penting untuk mitigasi risiko di wilayah pesisir dan aktivitas kelautan. Penelitian ini membandingkan efektivitas Grid Search dan Randomized Search dalam optimasi hyperparameter XGBoost untuk prediksi multioutput lima parameter cuaca dan laut, yaitu kecepatan angin, arah angin, kelembaban relatif, suhu permukaan laut, dan tinggi muka air menggunakan data Automatic Weather Station BMKG periode 2022-2025 pada stasiun Merak, Ciwandan, dan Bakauheni. Penelitian meliputi tahapan pengumpulan data, ekstraksi fitur temporal, pembersihan data, pemisahan fitur-target, pembagian data latih-uji, normalisasi, pelatihan model, hyperparameter tuning, dan evaluasi menggunakan metrik R². Hasil menunjukkan kedua metode menghasilkan akurasi tinggi dengan nilai R² berada pada rentang 0,7214 hingga 0,9916. Perbedaan akurasi antar metode relatif kecil, dengan selisih R² terbesar 0,0152 pada variabel arah angin. Meskipun demikian, Randomized Search menunjukkan keunggulan signifikan dari sisi efisiensi komputasi dengan waktu eksekusi hingga 7,5 kali lebih cepat dibandingkan Grid Search, yaitu 38,3 detik dibandingkan 285,6 detik pada variabel kecepatan angin. Selain itu, proses optimasi hyperparameter terbukti meningkatkan kinerja model dibandingkan konfigurasi baseline, dengan peningkatan nilai R² arah angin dari 0,6137 menjadi 0,7366. Berdasarkan keseimbangan antara akurasi dan efisiensi komputasi, Randomized Search direkomendasikan sebagai optimasi hyperparameter untuk model XGBoost multioutput pada data kompleks dan berdimensi tinggi.

Kata kunci— XGBoost, prediksi multioutput, hyperparameter tuning, Grid Search, Randomized Search

1. PENDAHULUAN

Dinamika atmosfer dan oseanografi yang terjadi di perairan laut dapat mengancam aktivitas wisata, keselamatan nelayan, serta memicu bencana di wilayah pesisir [1]. Prediksi variabel atmosfer-oseanografi memiliki peran penting dalam meminimalisir ancaman tersebut dengan menyediakan informasi yang cepat dan akurat. Pendekatan prediksi *multioutput* menjadi solusi yang mampu memprediksi beberapa variabel target sekaligus dari variabel input yang sama [2]. Metode ini telah banyak digunakan di berbagai bidang, termasuk meteorologi, oseanografi, kesehatan, dan sistem rekomendasi. Dalam konteks cuaca dan laut, prediksi *multioutput* memungkinkan pemodelan kondisi secara lebih menyeluruh, sehingga mendukung pengambilan keputusan yang cepat dan tepat. Penelitian ini menggunakan data dari *Automatic Weather Station* (AWS) sebagai studi kasus penerapan prediksi *multioutput*. AWS merupakan perangkat berbasis sensor elektrik yang berfungsi untuk mengukur parameter atmosfer dan oseanografi secara otomatis [3]. Pada penelitian ini, lima parameter penting digunakan sebagai variabel, yaitu kecepatan angin, arah angin, kelembaban relatif, suhu permukaan laut, dan tinggi muka air laut. Data tersebut memiliki potensi besar untuk dianalisis, namun karakteristiknya yang bersifat *multivariat*, berdimensi tinggi, dan memiliki hubungan *non-linier* antar variabel menimbulkan tantangan tersendiri dalam pemodelan. Oleh karena itu, dibutuhkan algoritma prediktif yang

mampu menangani interaksi kompleks antar variabel serta memproses volume data besar secara efisien, sehingga model dapat menghasilkan prediksi dengan akurasi tinggi dan aplikatif.

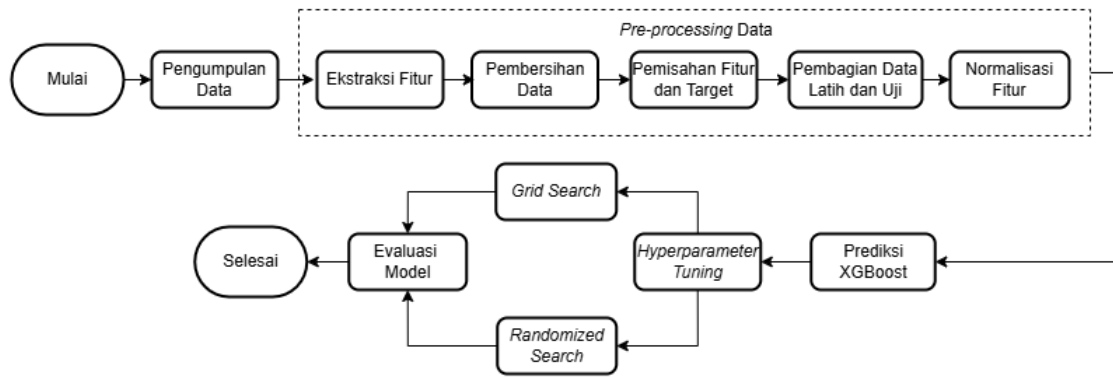
Extreme Gradient Boosting (XGBoost) dipilih sebagai algoritma dalam penelitian ini karena merupakan salah satu implementasi *gradient boosting* yang terbukti efisien dalam menangani data berskala besar dengan kinerja optimal [4]. XGBoost mengadopsi pendekatan *gradient boosting* yang menggabungkan sejumlah pohon keputusan secara bertahap untuk meningkatkan performa prediksi. Salah satu keunggulan XGBoost adalah efisiensi proses *training*, disertai kemampuan mengendalikan *overfitting* dengan teknik regularisasi yang lebih efektif [5]. Fleksibilitas algoritma ini juga memungkinkan penerapannya pada berbagai kasus, termasuk klasifikasi, regresi, hingga prediksi *multioutput*. Adaptasi XGBoost untuk prediksi *multioutput* memungkinkan kelima parameter cuaca dan laut dapat diprediksi secara bersamaan. Kinerja XGBoost sangat dipengaruhi oleh ketepatan pemilihan *hyperparameter*, yang berperan dalam menentukan konfigurasi pemodelan dan mengatur kompleksitas algoritma [6]. Proses pencarian kombinasi *hyperparameter* terbaik, atau *hyperparameter tuning*, menjadi langkah krusial untuk mengoptimalkan akurasi, generalisasi, dan efisiensi model. Dua pendekatan yang umum digunakan dalam *tuning* adalah *Grid Search* dan *Randomized Search*, masing-masing dengan karakteristik berbeda. *Grid Search* melakukan evaluasi sistematis terhadap seluruh kombinasi parameter dalam ruang pencarian dan menentukan konfigurasi optimal berdasarkan indikator evaluasi yang diperoleh melalui prosedur *cross-validation*, dengan data dibagi menjadi subset untuk pelatihan dan validasi secara bergantian [7]. Sebaliknya, *Randomized Search* merupakan metode optimasi *hyperparameter* yang mengeksplorasi kandidat secara acak dari ruang pencarian untuk menemukan konfigurasi terbaik [8]. Perbedaan karakteristik keduanya memunculkan pertanyaan terkait efektivitas dan efisiensi masing-masing metode dalam menghasilkan konfigurasi optimal, khususnya untuk XGBoost pada prediksi *multioutput*.

Penelitian yang secara langsung membandingkan *Grid Search* dan *Random Search* untuk optimasi *hyperparameter* pada XGBoost dalam kasus prediksi *multioutput* berbasis data AWS masih jarang dilakukan. Penelitian [9] telah membandingkan kedua metode tersebut pada algoritma XGBoost, namun berfokus pada klasifikasi penyakit gagal ginjal kronis dengan target *single-output*. Penelitian [10] juga menerapkan kedua metode *tuning* untuk klasifikasi penyakit diabetes, tetapi belum membahas penerapannya pada regresi berbasis data deret waktu (*time-series*). Sementara itu, penelitian [11] menggunakan XGBoost untuk memprediksi suhu udara maksimum dengan optimasi *hyperparameter*, namun hanya menggunakan metode *Grid Search* dan tetap berfokus pada satu variabel keluaran. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah dari batasan penelitian-penelitian tersebut dengan membandingkan kinerja *Grid Search* dan *Random Search* dalam menghasilkan model XGBoost optimal untuk memprediksi lima parameter cuaca dan laut secara bersamaan. Evaluasi dilakukan berdasarkan akurasi prediksi, efisiensi waktu komputasi, dan konsistensi hasil, sebagai rekomendasi untuk penelitian selanjutnya dalam memilih *hyperparameter tuning* yang sesuai untuk pemodelan *multioutput* di berbagai bidang.

2. METODE PENELITIAN

2.1. Alur Penelitian

Penelitian ini mencakup sembilan tahapan utama, yang pengumpulan data, ekstraksi fitur, pembersihan data, pemisahan fitur dan target, pembagian data menjadi latih dan uji, normalisasi fitur, pelatihan model, penyetelan *hyperparameter*, serta evaluasi model. Seluruh rangkaian tahapan tersebut divisualisasikan dalam alur penelitian pada Gambar 1.



Gambar 1. Alur Penelitian

2.2. Pengumpulan Data

Sumber data dalam penelitian ini berasal dari AWS milik Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). AWS tersebut secara otomatis merekam berbagai parameter meteorologi dan oseanografi, antara lain kecepatan serta arah angin, suhu dan ketinggian air, hingga kelembaban relatif. Pengumpulan data dilakukan secara berkala setiap satu jam, mencakup rentang waktu dari tahun 2022 sampai 2025. Data yang dihasilkan memiliki resolusi temporal yang tinggi, sehingga sangat mendukung analisis prediktif yang lebih mendalam dan akurat. Sampel dataset yang digunakan pada penelitian ini disajikan pada Tabel 1.

Tabel 1. Sampel Dataset AWS BMKG

no	waktu	lat	lon	winddir	windspeed	rh	watertemp	waterlevel
1	2022-01-01 00:00:00	-5.926459	105.886146	231.242 150	2.829114	86.5902 60	29.142172	3.152486
2	2022-01-01 00:00:00	-5.955795	105.937609	218.329 074	3.799139	87.9132 32	28.924046	2.335356
...
637 700 00	2025-02-06 23:58:00	-5.945715	105.905712	155.194 760	1.555569	81.4381 38	30.183579	1.648346

2.3. Pre-processing Data

2.3.1. Ekstraksi Fitur Waktu

Ekstraksi fitur merupakan tahap penting dalam pengolahan data karena berfungsi untuk memisahkan informasi relevan dari data berlebihan, sekaligus menyajikan representasi baru yang lebih ringkas untuk kebutuhan prediksi [12]. Melalui teknik ini, atribut awal diolah menjadi fitur yang lebih bermakna, informatif, dan relevan terhadap tujuan analisis [13]. Pada penelitian ini, ekstraksi difokuskan pada karakteristik temporal dari variabel waktu (timestamp) yang bersifat siklikal dan musiman, sehingga komponen seperti tahun, bulan, tanggal, dan jam dipisahkan menjadi fitur individual. Selain itu, digunakan representasi sinusoidal dan kosinusoidal pada bulan, hari, dan jam untuk menangkap pola periodik secara lebih baik, termasuk hubungan kedekatan antarwaktu. Fitur tambahan seperti weekday, is_weekend, dan season juga disertakan karena kondisi atmosfer dan laut dipengaruhi oleh variasi waktu harian, mingguan, dan musiman.

2.3.2. Pembersihan Data

Tahapan pembersihan data merupakan bagian penting dalam preprocessing karena berperan besar dalam peningkatan kualitas data dengan melalui imputasi nilai yang hilang serta

mengelminasi data yang tidak relevan [14]. Dalam penelitian ini, pembersihan data dilakukan dengan menghapus seluruh entri yang memiliki nilai kosong pada satu atau beberapa parameter.

2.3.3. Pemisahan Fitur dan Target

Setelah tahap pembersihan dan ekstraksi fitur dilakukan, data kemudian dipisahkan menjadi fitur dan target. Bagian fitur (X) terdiri dari berbagai parameter yang merepresentasikan kondisi lingkungan serta waktu, seperti koordinat geografis (*longitude* dan *latitude*), hasil transformasi waktu berbasis fungsi siklikal, dan tambahan atribut temporal seperti 'weekday', 'is_weekend', serta 'season'. Transformasi dan penambahan atribut temporal ini dilakukan untuk memfasilitasi model dalam mendeteksi hubungan antara data dan waktu secara lebih akurat [15]. Sementara itu, variabel target (y) berisi lima parameter utama terkait cuaca dan laut yang akan diprediksi secara bersamaan, yaitu kecepatan angin, tinggi muka air, suhu permukaan laut, kelembaban relatif, dan arah angin.

2.3.4. Pembagian data latih dan uji

Pembagian data merupakan tahapan pembagian dataset menjadi dua bagian, yaitu data latih dan data uji [16]. Pembagian ini dilakukan untuk mengevaluasi performa model secara lebih akurat dan objektif, dengan model dilatih menggunakan data latih untuk mempelajari pola antar variabel serta diuji pada data uji yang belum pernah dilihat sebelumnya [17]. Dalam penelitian ini, pembagian data dilakukan dengan proporsi 90% data latih dan 10% data uji, menggunakan fungsi *train test split* dari pustaka *scikit-learn* dengan *random state* yang disetel agar hasil pembagian bersifat *reproducible*.

2.3.5. Normalisasi Fitur

Normalisasi merupakan proses penting dalam pra-pemrosesan data untuk memastikan bahwa struktur data seragam dan dapat diinterpretasikan secara efektif oleh algoritma pembelajaran mesin [18]. Dalam konteks penelitian ini, normalisasi dilakukan untuk menyelaraskan skala setiap fitur numerik yang memiliki rentang nilai berbeda-beda. Ketidakseimbangan skala ini dapat menyebabkan fitur dengan nilai kecil menjadi kurang berpengaruh dalam proses pembelajaran dibandingkan fitur lain yang memiliki rentang lebih besar, sehingga dapat menurunkan kinerja model [19].

Dataset awal AWS BMKG yang digunakan terdiri dari 63.770.000 baris data dan 9 atribut. Setelah melalui tahapan preprocessing yang meliputi ekstraksi fitur waktu, pembersihan data, pemisahan fitur dan target, pembagian data latih dan uji, serta normalisasi fitur, jumlah data berkurang menjadi 22.132 baris dengan 22 atribut. Pengurangan jumlah data tersebut terjadi akibat proses pembersihan data yang menghapus entri dengan nilai kosong atau tidak lengkap pada beberapa parameter. Contoh dataset hasil preprocessing ditampilkan pada Tabel 2 dan Tabel 3 sebagai representasi data yang digunakan pada tahap pelatihan model.

Tabel 2. Contoh Dataset Hasil Pre-processing (Bagian 1)

waktu	lat	lon	wind_dir	wind_speed	rh	water_temp
2022-01-01 00:00	-5.93	105.89	227.98	3.17	87.35	29.04
2022-01-01 15:00	-5.93	105.89	225.96	4.86	86.27	28.61
2022-01-01 16:00	-5.93	105.89	238.95	4.45	87.98	28.62

Tabel 3. Contoh Dataset Hasil Pre-processing (Bagian 2)

water_level	month_sin	month_cos	hour_sin	hour_cos	weekday	is_weekend
2.91	0.50	0.86	0.00	1.00	5	1
2.40	0.50	0.86	-0.70	-0.70	5	1
2.38	0.50	0.86	-0.86	-0.50	5	1

2.4. Pembuatan Model

XGBoost merupakan pengembangan dari pohon keputusan yang menggunakan teknik *ensemble*, di mana model diperbarui secara bertahap untuk memperbaiki kesalahan yang terjadi pada model sebelumnya [20]. Proses pelatihan model dalam penelitian ini dilakukan dengan menggunakan algoritma XGBoost dalam kerangka regresi *multioutput*. Secara matematis, mekanisme kerja XGBoost dapat dijelaskan dengan persamaan berikut [21].

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in F \quad (1)$$

Keterangan:

- \hat{y}_i : Nilai prediksi untuk data ke- i
- K : Total jumlah pohon keputusan yang digunakan dalam model
- $f_k(x_i)$: Fungsi dari pohon regresi ke- k yang menghasilkan *output* berdasarkan input x_i
- F : Kumpulan semua kemungkinan fungsi pohon regresi yang dibentuk dalam model.

2.5. Hyperparameter Tuning

Hyperparameter tuning merupakan proses mencari kombinasi nilai parameter yang paling tepat untuk menghasilkan performa model yang maksimal. Melalui proses ini, model dapat menjadi lebih tangguh dalam menghadapi variasi data sekaligus menjaga konsistensi hasil prediksi [22]. *Grid Search* merupakan metode pencarian *hyperparameter* secara menyeluruh (*exhaustive search*) dengan menguji seluruh kombinasi nilai yang telah ditentukan sebelumnya untuk setiap *hyperparameter* [23]. Meskipun penerapannya cukup sederhana, *grid search* seringkali membutuhkan banyak waktu dan sumber daya dalam penerapannya [24]. *Randomized Search* adalah metode pencarian *hyperparameter* yang memilih kombinasi nilai secara acak dari ruang pencarian yang telah ditentukan [25]. Tidak seperti *Grid Search* yang menguji semua kemungkinan, metode ini hanya mencoba sejumlah kombinasi yang telah ditetapkan jumlah iterasinya (n_iter), sehingga dapat mengurangi waktu komputasi secara signifikan [26]. Meskipun tidak menjamin ditemukannya kombinasi terbaik mutlak, metode ini sering kali mampu menemukan konfigurasi yang mendekati optimal dengan biaya komputasi lebih efisien.

2.6. Evaluasi Model

Pada tahap evaluasi, penilaian difokuskan pada seberapa baik model XGBoost yang dibangun mampu memprediksi parameter-parameter cuaca dan laut secara akurat [27]. Evaluasi ini bertujuan untuk melihat kemampuan model dalam menggeneralisasi pola pada data uji, sehingga kita dapat memastikan model tidak hanya andal pada data latih, tetapi juga konsisten saat dihadapkan pada data baru. Evaluasi dilakukan menggunakan koefisien determinasi (R^2), yang menunjukkan seberapa besar variasi data dapat dijelaskan oleh model, di mana nilai yang semakin mendekati 1 menandakan tingkat akurasi prediksi yang lebih tinggi [28]. Perhitungan R^2 dilakukan berdasarkan Persamaan (2) berikut [29]:

$$R^2 = 1 - \frac{\sum_{i=1}^m (X_i - Y_i)^2}{\sum_{i=1}^m (\bar{Y} - Y_i)^2} \quad (2)$$

3. HASIL DAN PEMBAHASAN

3.1. Hasil Pengujian Model Sebelum Tuning

Sebelum dilakukan proses optimasi *hyperparameter*, model XGBoost dievaluasi menggunakan konfigurasi parameter default untuk menetapkan kinerja dasar (*baseline performance*). Pengujian ini untuk mengukur kemampuan prediktif model tanpa penyesuaian khusus, sekaligus menjadi acuan dalam menilai efektivitas tuning pada tahap berikutnya. Evaluasi kinerja pada data uji menggunakan metrik R^2 menunjukkan bahwa model memiliki performa bervariasi di antara parameter target. Hasil pengujian model ditampilkan pada Tabel 4.

Tabel 4. Kinerja Model XGBoost *Default* (Sebelum *Tuning*) pada Data Uji

Variabel Target	R ²
Kecepatan Angin (windspeed)	0.7582
Tinggi Muka Air (waterlevel)	0.8815
Suhu Permukaan Laut (watertemp)	0.9790
Kelembaban Relatif (rh)	0.7844
Arah Angin (winddir)	0.6137

Variabel suhu permukaan laut (watertemp) mencapai nilai R² tertinggi, yaitu 0.9790, menandakan kemampuan model yang sangat baik dalam menjelaskan variasi data. Variabel tinggi muka air (waterlevel), kelembaban relatif (rh), dan kecepatan angin (windspeed) juga menunjukkan kinerja yang cukup baik dengan nilai R² masing-masing sebesar 0.8815, 0.7844, dan 0.7582. Namun, untuk variabel arah angin (winddir), nilai R² yang diperoleh hanya sebesar 0.6137, menunjukkan tingkat akurasi prediksi yang lebih rendah akibat sifatnya yang lebih stokastik. Secara keseluruhan, hasil ini mengindikasikan bahwa meskipun model XGBoost dengan parameter *default* telah memberikan performa yang memadai untuk sebagian besar variabel, masih terdapat ruang yang signifikan untuk peningkatan, khususnya pada variabel dengan pola data yang lebih kompleks, melalui penerapan optimasi *hyperparameter*.

3.2. Hasil Optimasi Hyperparameter

Ruang pencarian (*search space*) untuk masing-masing metode dirancang berdasarkan parameter-parameter kunci yang memengaruhi performa model, meliputi *n_estimators*, *max_depth*, *learning_rate*, *subsample*, *colsample_bytree*, *reg_alpha*, dan *reg_lambda*. Pada *Grid Search*, setiap parameter memiliki daftar nilai diskrit yang telah ditentukan, sedangkan pada *Randomized Search* digunakan rentang nilai yang lebih luas untuk memungkinkan eksplorasi yang lebih fleksibel. Rincian ruang pencarian *hyperparameter* ditunjukkan pada Tabel 5.

Tabel 5. Ruang Pencarian Nilai *Hyperparameter*

Parameter	Ruang Pencarian	
	<i>Grid Search</i>	<i>Randomized Search</i>
<i>n_estimators</i>	1000, 1200, 1500	100, 300, 500, 700, 1000, 1200, 1500
<i>max_depth</i>	4, 6, 8	3, 4, 5, 6, 7, 8, 9, 10
<i>learning_rate</i>	0.01, 0.05, 0.1	0.01, 0.03, 0.05, 0.07, 0.1, 0.15, 0.2
<i>subsample</i>	0.8, 0.9, 1.0	0.7, 0.8, 0.85, 0.9, 0.95, 1.0
<i>colsample_bytree</i>	0.8, 0.9, 1.0	0.7, 0.8, 0.85, 0.9, 0.95, 1.0
<i>reg_alpha</i>	0, 0.1, 0.5	0, 0.1, 0.3, 0.5, 0.7, 1.0
<i>reg_lambda</i>	1, 2, 3	0.5, 1.0, 1.5, 2.0, 2.5, 3.0

Setelah proses pencarian selesai, kedua metode menghasilkan konfigurasi hyperparameter optimal yang berbeda. Hasil terbaik dipilih berdasarkan skor rata-rata R² tertinggi pada validasi silang (*cross-validation*). Tabel 6 menampilkan konfigurasi *hyperparameter* optimal yang diperoleh masing-masing metode.

Tabel 6. Nilai Parameter Terbaik Setiap Metode

Parameter	Nilai Terbaik	
	<i>Grid Search</i>	<i>Randomized Search</i>
<i>n_estimators</i>	1500	1500
<i>max_depth</i>	8	10
<i>learning_rate</i>	0.1	0.05

subsample	0.9	0.9
colsample_bytree	1.0	0.95
reg_alpha	0.5	0.5
reg_lambda	1	0.5

Grid Search menghasilkan kombinasi terbaik dengan *max depth* bernilai 8 dan *learning rate* bernilai 0.1, sedangkan *Randomized Search* menemukan kombinasi *max depth* bernilai 10 dan *learning rate* bernilai 0.05, yang menunjukkan perbedaan signifikan pada nilai kedalaman pohon dan laju pembelajaran. Parameter *n estimators* yang tinggi pada kedua metode menunjukkan kebutuhan model akan jumlah pohon keputusan yang banyak guna mencapai prediksi yang stabil dan akurat. Hal ini sejalan dengan yang disampaikan oleh [30], bahwa peningkatan jumlah *n estimators* dapat memperbaiki performa model melalui lebih banyak iterasi boosting. Selanjutnya, nilai *max depth* yang relatif besar pada hasil *Randomized Search* menandakan model mampu mempelajari pola data yang lebih kompleks, sebagaimana dijelaskan oleh Sudarman dan Budi [31], di mana struktur pohon yang lebih dalam memungkinkan penangkapan pola yang lebih rumit. Untuk mengatasi risiko *overfitting* akibat tingginya kompleksitas model, diterapkan regularisasi melalui parameter *reg alpha* dan *reg lambda*, yang berfungsi membatasi bobot model sehingga tetap generalis [32]. Secara umum, hasil ini mengindikasikan bahwa *Randomized Search* memiliki kemampuan menjangkau kombinasi nilai yang tidak tercakup dalam *Grid Search*. Pendekatan ini memberikan peluang lebih besar untuk menemukan konfigurasi optimal, khususnya ketika ruang pencarian sangat luas.

3.3. Perbandingan Hasil Tuning Hyperparameter

Berdasarkan Tabel 7, perbandingan performa antara metode *Grid Search* dan *Randomized Search* menunjukkan bahwa kedua pendekatan mampu menghasilkan nilai R^2 yang tinggi dan kompetitif pada seluruh variabel target. Perbedaan skor R^2 antar metode relatif kecil, yang menandakan bahwa baik *Grid Search* maupun *Randomized Search* sama-sama mampu mengidentifikasi kombinasi *hyperparameter* yang berkinerja baik. Meski demikian, *Randomized Search* secara konsisten unggul pada sebagian besar variabel, terutama pada *windspeed*, *rh*, dan *winddir*, dengan selisih terbesar mencapai 0.0152 pada *winddir*. Sementara itu, *Grid Search* hanya lebih unggul pada variabel *waterlevel*, meskipun selisihnya tergolong kecil (-0.0056).

Tabel 7. Hasil perbandingan metode grid search dan random search

Variabel	R^2		Waktu (s)	
	<i>Grid Search</i>	<i>Random Search</i>	<i>Grid Search</i>	<i>Random Search</i>
<i>windspeed</i>	0.8317	0.8411	285.6	38.3
<i>waterlevel</i>	0.9455	0.9399	267.2	27.3
<i>watertemp</i>	0.9913	0.9916	260.7	18.5
<i>rh</i>	0.8647	0.8730	280.3	40.3
<i>winddir</i>	0.7214	0.7366	279.6	40.7

Selain membandingkan akurasi prediksi, efisiensi waktu komputasi dari kedua metode *tuning* juga diperhatikan. Dari Tabel 4 dapat dilihat bahwa *Grid Search* membutuhkan waktu yang jauh lebih lama dibandingkan *Randomized Search* pada setiap variabel target. Sebagai ilustrasi, untuk parameter kecepatan angin (*windspeed*), proses tuning dengan *Grid Search* memakan waktu sekitar 285,6 detik, sedangkan *Randomized Search* hanya memerlukan 38,3 detik. Pola serupa juga terlihat untuk variabel lain, di mana *Randomized Search* secara konsisten menyelesaikan proses *tuning* lebih cepat bahkan selisih waktu bisa mencapai sepuluh kali lipat.

Perbedaan waktu ini terutama disebabkan oleh perbedaan mekanisme pencarian pada kedua metode. *Grid Search* secara sistematis menguji setiap kombinasi *hyperparameter* yang

mungkin, sehingga jumlah eksperimen menjadi sangat banyak, apalagi jika ruang pencarian luas. Sebaliknya, *Randomized Search* mengambil sampel acak dari seluruh ruang pencarian, sehingga mampu mengeksplorasi berbagai kombinasi secara efisien tanpa mencoba semuanya satu per satu.

4. KESIMPULAN

Penelitian ini berhasil membangun model prediksi multioutput berbasis XGBoost untuk memperkirakan parameter cuaca dan laut seperti kecepatan angin, arah angin, kelembaban relatif, suhu permukaan laut, dan tinggi muka air menggunakan data AWS BMKG tahun 2022–2025. Hasil pengujian model default menunjukkan kinerja yang cukup baik dengan nilai R^2 tertinggi sebesar 0,9790 pada variabel suhu permukaan laut dan nilai terendah sebesar 0,6137 pada arah angin. Penerapan hyperparameter tuning melalui *Grid Search* dan *Randomized Search* berhasil meningkatkan performa model secara keseluruhan. Metode *Randomized Search* menunjukkan keunggulan pada sebagian besar variabel target dengan selisih R^2 hingga 0,0152 dibandingkan *Grid Search*, serta memberikan efisiensi waktu komputasi hingga 10 kali lebih cepat. Hal ini membuktikan bahwa *Randomized Search* lebih efektif untuk ruang pencarian parameter yang luas, sedangkan *Grid Search* tetap unggul untuk pencarian yang sistematis.

5. SARAN

Penelitian selanjutnya disarankan untuk menambahkan parameter oseanografi tambahan seperti arus laut, salinitas, dan klorofil-a guna meningkatkan akurasi serta kemampuan generalisasi model. Selain itu, perlu dilakukan pengembangan metode optimasi lanjutan seperti *Bayesian Optimization* atau algoritma genetika untuk memperoleh hasil *tuning* yang lebih efisien serta membandingkan performa XGBoost dengan model pembelajaran mendalam seperti LSTM.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) atas penyediaan data yang digunakan dalam penelitian ini, serta kepada Universitas Pendidikan Indonesia Kampus Serang atas dukungan akademik dan fasilitas penelitian. Penulis juga berterima kasih kepada pihak-pihak yang telah memberikan dukungan, baik secara moral maupun material, sehingga penelitian ini dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] N. N. B. Marscelina, I. G. L. Wijayakusuma, and P. V. Swastika, “Perbandingan Metode LSTM dan TCN untuk Prediksi Gelombang Laut Berdasarkan Enam Parameter Oseanografi,” *JST (Jurnal Sains dan Teknol.*, vol. 14, no. 1, pp. 56–66, May 2025, doi: 10.23887/jstundiksha.v14i1.92590.
- [2] S. Emami and G. Martinez-Munoz, “Deep Learning for Multi-Output Regression Using Gradient Boosting,” *IEEE Access*, vol. 12, pp. 17760–17772, 2024, doi: 10.1109/ACCESS.2024.3359115.
- [3] H. S. Wicaksana, M. Putra, and D. P. Djenal, “Evaluasi Kinerja Automatic Weather Station Berdasarkan Pengamatan Paralel di Stasiun Meteorologi Kemayoran,” in *Prosiding Seminar Nasional Teknik Elektro*, 2021.
- [4] Z. Mustaffa and M. H. Sulaiman, “Advanced forecasting of building energy loads with XGBoost and metaheuristic algorithms integration,” *Energy Storage Sav.*, Aug. 2025, doi: 10.1016/j.enss.2025.03.005.
- [5] I. Maulita and A. M. Wahid, “Prediksi Magnitudo Gempa Menggunakan Random Forest, Support Vector Regression, XGBoost, LightGBM, dan Multi-Layer Perceptron Berdasarkan Data Kedalaman

- dan Geolokasi,” *J. Pendidik. dan Teknol. Indones.*, vol. 4, no. 5, pp. 221–232, May 2024, doi: 10.52436/1.jpti.470.
- [6] J. A. Ilemobayo *et al.*, “Hyperparameter Tuning in Machine Learning: A Comprehensive Review,” *J. Eng. Res. Reports*, vol. 26, no. 6, pp. 388–395, Jun. 2024, doi: 10.9734/jerr/2024/v26i61188.
- [7] Y. Zhao, W. Zhang, and X. Liu, “Grid search with a weighted error function: Hyper-parameter optimization for financial time series forecasting,” *Appl. Soft Comput.*, vol. 154, Mar. 2024, doi: 10.1016/j.asoc.2024.111362.
- [8] S. A. Tiastama and I. Budi, “Perbandingan Random Search dan Algoritma Genetika dalam Penyetelan Hyperparameter XGBoost pada Retail Sales Forecasting,” *Indones. J. Comput. Sci.*, vol. 13, no. 4, Aug. 2024, doi: 10.33022/ijcs.v13i4.4285.
- [9] D. A. Anggoro and S. S. Mukti, “Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure,” *Int. J. Intell. Eng. Syst.*, vol. 14, no. 6, pp. 198–207, 2021.
- [10] G. Abdurrahman, H. Oktavianto, and M. Sintawati, “Optimasi Algoritma XGBoost Classifier Menggunakan Hyperparameter GridSearch dan Random Search pada Klasifikasi Penyakit Diabetes,” *INFORMAL: Informatics J.*, vol. 7, no. 3, pp. 193–198, 2022.
- [11] S. Sugiarto, I. G. S. M. Diyasa, D. S. Alhamda, R. L. Aryananda, A. R. F. Sari, H. Sukri, and D. A. Dewi, “Optimizing The XGBoost Model with Grid Search Hyperparameter Tuning for Maximum Temperature Forecasting,” *J. Appl. Data Sci.*, vol. 6, no. 4, pp. 2517–2529, 2025.
- [12] W. Jia, M. Sun, J. Lian, and S. Hou, “Feature dimensionality reduction: a review,” *Complex Intell. Syst.*, vol. 8, no. 3, pp. 2663–2693, Jun. 2022, doi: 10.1007/s40747-021-00637-x.
- [13] D. Rohadi, “Skripsi Pengaruh Ensemble Feature Selection pada Prediksi Data Time Series Menggunakan Gated Recurrent Unit (GRU) dan Bidirectional Long Short-Term Memory (Bi-LSTM) (Studi Kasus: Walmart) Diususun oleh,” UIN Syarif Hidayatullah Jakarta, 2024.
- [14] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, “A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data,” *Front. Energy Res.*, vol. 9, Mar. 2021, doi: 10.3389/fenrg.2021.652801.
- [15] F. J. López-Andreu, J. A. López-Morales, Z. Hernández-Guillen, J. A. Carrero-Rodrigo, M. Sánchez-Alcaraz, J. F. Atenza-Juárez, and M. Erena, “Deep learning-based time series forecasting models evaluation for the forecast of chlorophyll a and dissolved oxygen in the Mar Menor,” *J. Mar. Sci. Eng.*, vol. 11, no. 7, July 2023, art. no. 1473, doi: [10.3390/jmse11071473](https://doi.org/10.3390/jmse11071473).
- [16] Q. H. Nguyen *et al.*, “Influence of data splitting on performance of machine learning models in prediction of shear strength of soil,” *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/4832864.
- [17] S. Prayudani, Y. Sibarani, A. Salam, and A. R. Lubis, “Perbandingan Kinerja Model Pembelajaran Mesin Random Forest dan K-Nearest Neighbor (KNN) untuk Prediksi Risiko Kredit pada Layanan Pinjaman Online,” *J. Software, Hardw. Inf. Technol.*, vol. 5, no. 2, pp. 118–127, Jun. 2025, doi: 10.24252/shift.v5i2.204.
- [18] H. Karmila and S. Yuliyatini, “Systematic Literature Review: Effective Data Normalization Method in Detecting Diabetes Using Machine Learning (K-Nearest Neighbors Algorithm),” *J. Sci. Res. Dev.*, vol. 6, no. 1, 2024, [Online]. Available: <https://idm.or.id/JSCR/inde>
- [19] P. P. Allorerung, A. Erna, M. Bagussahrir, and S. Alam, “Analisis Performa Normalisasi Data untuk

- Klasifikasi K-Nearest Neighbor pada Dataset Penyakit,” *J. Inform. Sunan Kalijaga*), vol. 9, no. 3, pp. 178–191, Sep. 2024.
- [20] F. I. Sari, E. L. Gunawan, C. A. Adhigiadany, and A. Lisanthoni, “Model Prediksi Kepadatan Lalu Lintas: Perbandingan Antara Algoritma Random Forest dan XGBoost,” *Semin. Nas. Sains Data*, vol. 2023, 2023.
- [21] R. Winurputra and D. E. Ratnawati, “Peramalan Penjualan Produk Menggunakan Extreme Gradient Boosting (XGBoost) dan Kerangka Kerja CRISP-DM untuk Pengoptimalan Manajemen Persediaan (Studi Kasus: UB Mart),” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 12, no. 2, pp. 417–428, Apr. 2025, doi: 10.25126/jtiik.2025129451.
- [22] A. Amalia, M. Radhi, D. R. H. Sitompul, S. H. Sinurat, and E. Indra, “Prediksi Harga Mobil Menggunakan Algoritma Regresi dengan Hyper-Parameter Tuning,” *J. Sist. Inf. dan Ilmu Komput. Prima*, vol. 4, no. 2, Feb. 2021.
- [23] A. D. Rachmatsyah, T. Sugihartono, and K. Irfan, “Perbandingan Teknik Optimasi Grid Search dan Randomized Search dalam Meningkatkan Akurasi Metode Klasifikasi SVM Pada Sentimen Ulasan Pengguna Aplikasi JKN Mobile,” *SKANIKA Sist. Komput. dan Tek. Inform.*, vol. 8, no. 1, pp. 13–22, Jan. 2025.
- [24] S. R. Sidiq and A. Salam, “SciBERT Optimisation for Named Entity Recognition on NCBI Disease Corpus with Hyperparameter Tuning,” *J. Appl. Informatics Comput.*, vol. 9, no. 2, p. 432, Apr. 2025, [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [25] M. Fajri and A. Primajaya, “Komparasi Teknik Hyperparameter Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan Grid Search dan Random Search,” *J. Appl. Informatics Comput.*, vol. 7, no. 1, pp. 2548–6861, Jul. 2023, [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [26] H. Mulyo and A. K. Zyen, “Pengaruh Hyperparameter Tuning Gradient Boosting Terhadap Prediksi Pemilihan Program Studi Mahasiswa Baru,” *Bull. Comput. Sci. Res.*, vol. 5, no. 2, pp. 131–137, Feb. 2025, doi: 10.47065/bulletincsr.v5i2.454.
- [27] M. F. R. Aditya, N. L. Azizah, and U. Indahyanti, “Prediksi Penyakit Hipertensi Menggunakan Metode Decision Tree dan Random Forest,” *J. Ilm. Komputasi*, vol. 23, no. 1, Mar. 2024, doi: 10.32409/jikstik.23.1.3503.
- [28] A. B. Fawait, M. Jamil, S. Rahmah, and S. Sugiarto, “Penerapan Metode LSTM untuk Prediksi Harga Ethereum,” *J. Rekayasa Teknol. Inf.*, vol. 9, no. 3, pp. 243–252, 2025.
- [29] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Comput. Sci.*, vol. 7, July 2021, art. no. e623, doi: [10.7717/peerj-cs.623](https://doi.org/10.7717/peerj-cs.623)
- [30] G. Chairunisa *et al.*, “Life Expectancy Prediction Using Decision Tree, Random Forest, Gradient Boosting, and XGBoost Regressions,” *J. Sintak*, vol. 2, no. 2, Mar. 2024.
- [31] E. J. Sudarman and S. Budi, “Pengembangan Model Kecerdasan Mesin Extreme Gradient Boosting untuk Prediksi Keberhasilan Studi Mahasiswa,” *J. Strateg.*, vol. 5, Nov. 2023.
- [32] M. B. Ilmi and K. Kusriani, “Perbandingan Kinerja Algoritma Machine Learning dalam Deteksi Potensi Risiko HIV,” *J. BUFFER Inform.*, vol. 11, no. 1, Apr. 2025, [Online]. Available: <https://journal.fkom.uniku.ac.id/buffer>