Prediksi Rain Sum dengan Menggunakan Pembelajaran Mesin Berbasis *Gradient Boosted Tree*

Andika Zidane Fathurrahman

Muhammad Pudja Gemilang

Muhhammad Nabil Fadhlurrahman

Ayam Girang

Datavidia 2023

DAFTAR PUSTAKA

I. PENDAHULUAN

1.1. Latar Belakang

Prakiraan cuaca secara umum merupakan penggunaan ilmu dan teknologi untuk memprakirakan keadaan cuaca pada masa mendatang pada suatu tempat. Seiring dengan perkembangan pembelajaran mesin yang sangat pesat, masalah prakiraan cuaca dapat ditransformasi menjadi permasalah prediksi pada pembelajaran mesin.

Gradient boosted machine learning merupakan model *ensamble* berbasis *decision tree* yang terdiri dari model-model lemah yang dioptimisasikan sehingga diperoleh model yang lebih baik dalam memberikan prediksi secara *robust*. Model *gradient boosted machine learning* yang sangat populer digunakan dalam menyelesaikan masalah regresi dan klasifikasi adalah model *Extreme Gradient Boosting* (XGBoost) dan *Light Gradient Based Machine* (LGBM).

Pada kompetisi ini, akan dilakukan prediksi curah hujan dalam milimeter pada 10 kota di masa mendatang jika diberikan data historis berupa temperatur, kecepatan angin, tekanan udara, shortwave radiation, dan fitur lainnya.

1.2. Rumusan Masalah

1. Bagaimana performa model *gradient boosted machine learning* dalam memprediksi *rain sum* dari setiap kota

1.3. Tujuan Penelitian

1. Untuk menentukan performa model *gradient boosted machine learning* dalam memprediksi *rain sum* dari setiap kota

II. PENGOLAHAN DATA

2.1. Interpolasi Data

Data *train* dan *test* memiliki beberapa fitur yang mengandung nilai kosong. Adapun persentase besar nilai kosong yang terdapat di data *train* dan *test* adalah sebagai berikut,

Tabel II-1 Persentase missing values pada data train dan test

No.	Kolom	Persentase (%)	
		train	*test*
1.	time	0	0
2.	temperature_2m_max (°C)	0.377131	0
3.	temperature_2m_min (°C)	0.377131	0
4.	apparent_temperature_max (°C)	0.377131	0
5.	apparent_temperature_min (°C)	0.377131	0
6.	sunrise (iso8601)	0	0
7.	sunset (iso8601)	0	0
8.	shortwave_radiation_sum (MJ/m²)	0.452557	0
9.	rain_sum (mm)	0.452557	-
10.	snowfall_sum (cm)	0.452557	0
11.	windspeed_10m_max (km/h)	0.377131	0
12.	windgusts_10m_max (km/h)	0.377131	0
13.	winddirection_10m_dominant (°)	3.514859	0
14.	et0_fao_evapotranspiration (mm)	0.452557	0
15.	elevation	0	0
16.	city	0	0

Perhatikan bahwa ada 11 fitur di data *train* yang memiliki nilai kosong yang sedikit dan pada data *test* tidak ada fitur yang memiliki nilai kosong.

Langkah pertama untuk menangani nilai kosong dalam fitur *rain_sum* adalah untuk membuang entri yang mengandung nilai kosong di *rain_sum*. Selanjutnya untuk mengisi nilai kosong di fitur lainnya, digunakan metode interpolasi linier dengan kedua titik observasi yang terpisah oleh beberapa nilai kosong dihubungkan oleh suatu garis linear. Hal

ini ditujukan untuk mengikuti sifat transisi cuaca secara perlahan dari hari ke hari. Sebagai ilustrasi, jika diberikan hari ke-i hujan, maka kemungkinan besar hari ke-$(i+1)$ juga akan hujan.

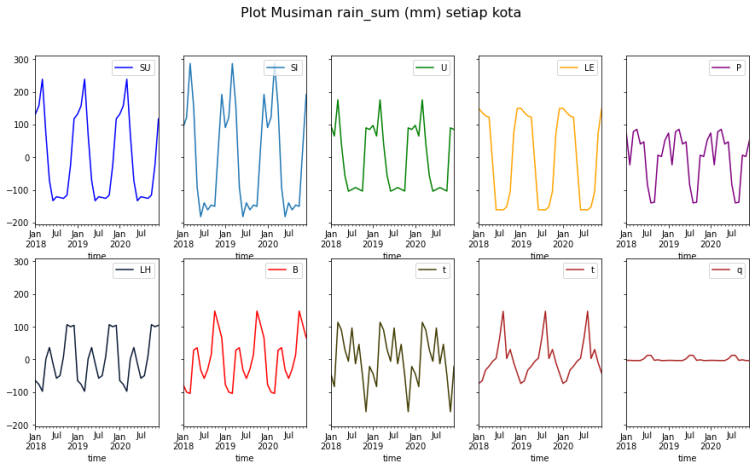
2.2. Mengagregatkan Data *Hourly* menjadi Data per Hari

Pada bagian *feature engineering*, akan digunakan informasi yang terkandung di data *train* dan *test hourly*. Adapun informasi yang akan digunakan dari kedua data tersebut adalah semua fitur yang tidak terkandung di data *train* dan *test*. Untuk mengagregatkan observasi per jam dari suatu hari menjadi observasi per hari, dihitung besar rata-rata, nilai maksimum, dan standar deviasi pada suatu hari dari setiap informasi yang ingin diperoleh. Dengan mengagregatkan data *hourly*, informasi dari data *hourly* dapat digabungkan dengan data *train* dan *test* untuk pemodelan selanjutnya.

III. Eksplorasi Analisis Data

3.1. Plot Musiman Bulanan dari rain sum (mm) per Kota

Berikut disajikan plot musiman per bulan dari setiap kota,



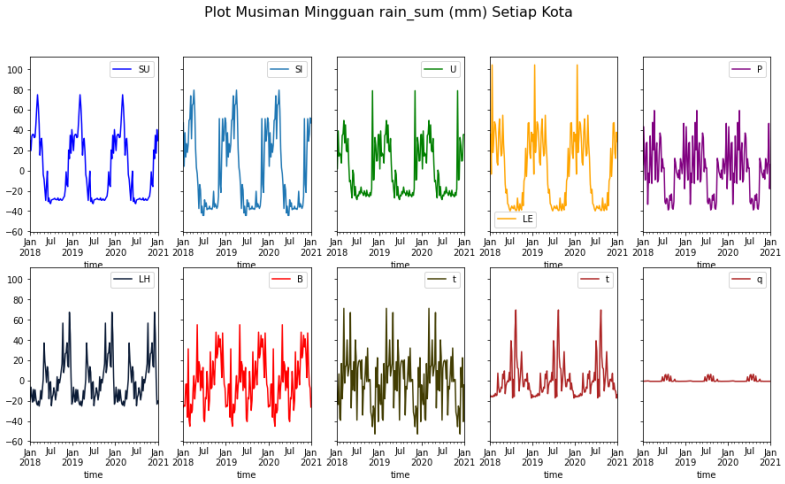
Gambar III-1 Plot musiman per bulan rain_sum (mm)

Diatas merupakan plot musiman dari rain sum (mm) per kota. Dapat diamati bahwa untuk rain sum di setiap kota memiliki pola musiman. Lebih lanjut, untuk setiap kotanya memiliki pola musiman yang berbeda. Walaupun tingkat intensitas hujan yang berbeda disetiap kotanya, terdapat bulan tertentu yang memiliki intensitas hujan yang lebih tinggi. Jika diamati pada

plot gambar, setiap kota kecuali kota q, intensitas hujannya tinggi pada bulan November sampai Mei daripada bulan yang lain.

3.2. Plot Musiman Mingguan dari rain sum (mm) per Kota

Berikut disajikan plot musiman per minggu dari setiap kota

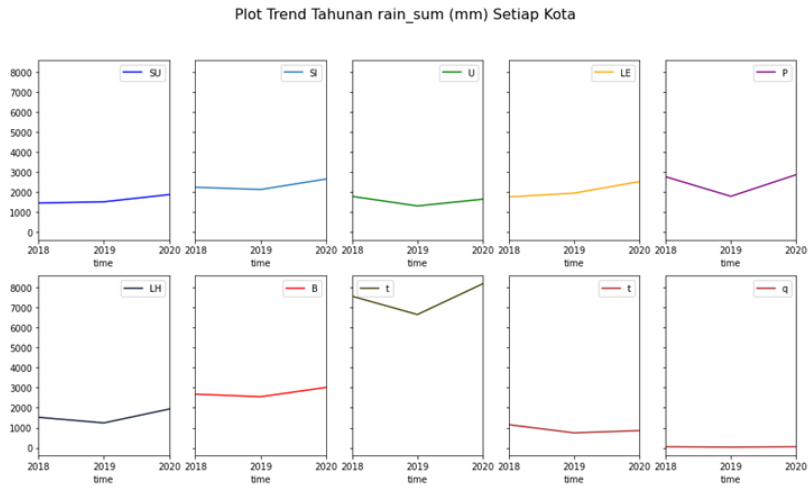


Gambar III-2 Plot musiman per minggu rain sum

Diatas merupakan plot musiman per minggu dari *rain sum* per kota. Dapat diamati bahwa datanya lebih *noisy* daripada data per bulan. Walaupun terlihat lebih acak, plot musiman per minggu dari *rain sum* per kota ini masih memiliki pola yang mirip seperti data musiman bulanannya. Dengan meningkatnya presisi plot dari awalnya hanya musiman bulanan ke musiman per minggu, kita mendapatkan terdapat minggu-minggu tertentu yang meiliki intensitas tinggi dan berlaku musiman.

3.3. Plot Trend Tahunan dari rain sum (mm) per kota

Berikut disajikan plot trend tahunan dari setiap kota.

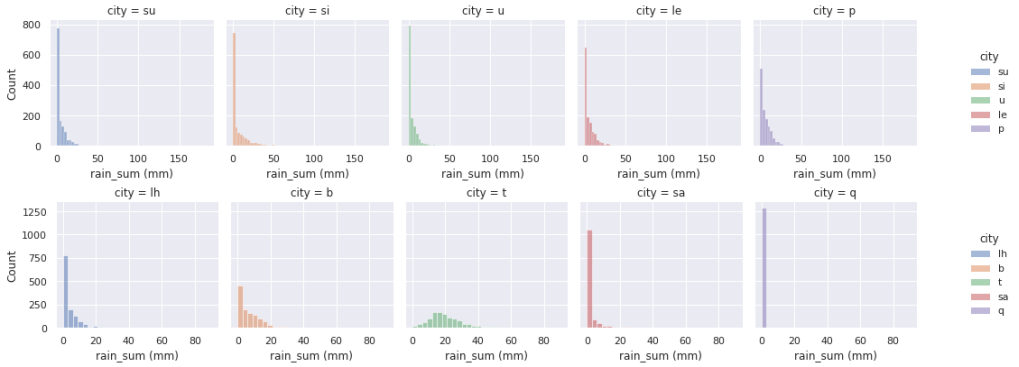


Gambar III-3 Plot trend tahunan rain sum

Di atas merupakan plot trend tahunan dari *rain sum* per kota. Dapat dilihat bahwa untuk setiap kota memiliki pola yang sama yaitu, *rain sum* yang menurun pada tahun 2018-2019 dan meningkat lagi pada tahun 2019-2020. Pola seperti itu dapat terlihat dengan jelas kecuali pada kota q yang memiliki trend terbilang tidak turun maupun naik atau statis. Dari plot juga dapat terlihat bahwa rain sum terbanyak pada tahun 2018-2020 ada pada kota *t*.

3.4. Distribusi *rain sum* per Kota

Berikut disajikan plot distribusi *rain sum* untuk setiap kota



Gambar III-4 Distribusi rain sum per kota

Dapat diamati diatas merupakan plot distribusi *rain sum* (mm) per kota. Terlihat jelas bahwa semua kota memiliki bentuk *right skewed* selain pada kota q. Dengan demikian, pada kota selain kota q, rerata *rain sum* jauh lebih besar daripada nilai median dan juga nilai modusnya. Hal tersebut dapat mempengaruhi proses pemodelan yang nanti akan dilakukan.

Sedangkan pada kota q, nilai *rain sum* (mm) tidak jauh dari 0 dan memiliki distribusi disekitar 0.

3.5. Uji Kestationeran

Sebelum melakukan pemodelan, akan dilakukan uji diagnostik terlebih dahulu. Uji yang akan dilakukan adalah uji kestationeran terhadap variable rain sum (mm). Digunakan test *Augmented Dickey-Fuller* (ADF) untuk menguji kestasioneran. Pertama akan dipilih $\alpha = 0.05$ dengan uji hipotesis sebagai berikut,

$$H_0: \text{Data tidak stasioner}$$
$$H_1: \text{Data stasioner}$$

Jika $p - value$ lebih kecil atau sama dengan α, maka H_0 ditolak, sehingga dapat simpulkan bahwa datanya stasioner, begitu pula sebaliknya. Berikut merupakan hasil percobaan pada data *train*,

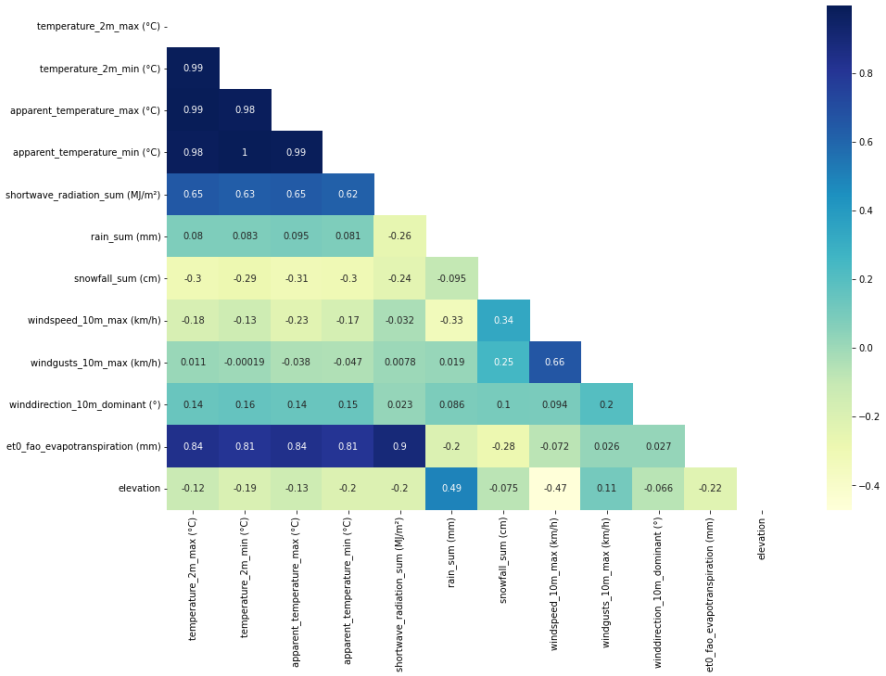
Tabel III-1 Nilai p-value untuk rain sum di setiap kota

No.	Kota	$p - value$
1.	Su	0.002173
2.	Si	0.000889
3.	U	0.0002
4.	Le	0.01
5.	P	0.00002
6.	Lh	3×10^{-5}
7.	B	4×10^{-17}
8.	T	5.8×10^{-23}
9.	Sa	1.18×10^{-17}
10.	Q	12×10^{-6}

Karena $p - value$ dari uji ADF-test masing-masing kota kurang dari α yang dipilih, yaitu 0.05, maka tidak punya cukup bukti untuk menunjukkan bahwa data tidak stasioner untuk *rain sum* setiap kotanya.

3.6. Analisis Multivariat

Berikut disajikan *heatmap* dari korelasi antar fitur



Gambar III-5 Heatmap korelasi antar fitur

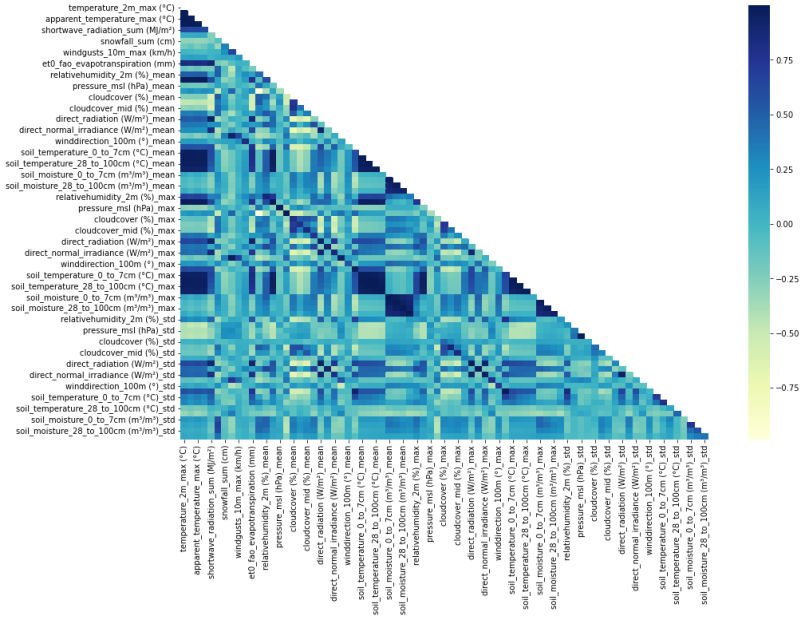
Di atas merupakan plot *heatmap* yang menyatakan korelasi *pearson* antar variabel. Jika dipilih variabel terikatnya adalah *rain sum* dan yang lain merupakan variabel bebas, maka variabel *elevation* merupakan variabel yang memiliki korelasi positif terbesar terhadap *rain sum*. Didapatkan pula variabel *windspeed_10m_max* (km/h) memiliki korelasi negatif terbesar terhadap *rain sum*. Namun, antar variabel temperature memiliki korelasi yang sangat tinggi. Dengan demikian, terdapat masalah multikolinearitas pada data ini.

IV. REKAYASA FITUR

4.1. Penggabungan Data Hourly dengan Data Train/Test

Setelah dilakukan pengagregatan data *hourly* menjadi data per hari, informasi terkait dengan rerata, nilai maksimum dan standar deviasi dari setiap informasi yang eksklusif di data *hourly* digabungkan dengan data *train* dan *test*.

Penggabungan data *hourly* dengan data *train* dan *test* menghasilkan 66 fitur baru untuk dilatih dengan model *machine learning* beserta fitur yang terdapat di data *train* dan *test*. Berikut disajikan diagram *heatmap* dari penggabungan data *hourly* dengan data *train*



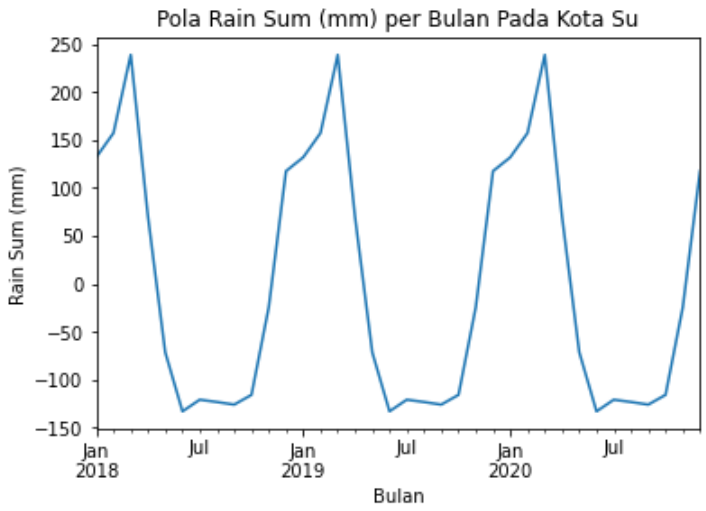
Gambar IV-1 Diagram heatmap korelasi antar fitur

Dapat dilihat dari diagram tersebut bahwa beberapa fitur multikolinear antar sesamanya.

4.2. Ekstrak Fitur Tanggal

Karena model *machine learning* tidak dapat memproses informasi berupa tanggal, maka informasi terkait tanggal diekstrak dalam bentuk numerik. Pada penelitian ini digunakan informasi minggu, bulan, dan tahun dalam bentuk angka. Selanjutnya, dibuatkan fitur biner baru bernama *rainy_month* terkait dengan kemunculan musim hujan. Observasi musim hujan diamati dari pola/trend variabel *rain_sum* (mm) dari hari ke hari.

Untuk ilustrasi pola/trendnya dapat dilihat dari grafik *rain_sum* (mm) dari kota Su sebagai berikut,



Gambar IV-2 Grafik garis rain sum (mm) per bulan di Kota Su

Dapat diamati dari grafik tersebut bahwa, intensitas *rain sum* yang tinggi terjadi di bulan November sampai bulan Mei. Kami asumsikan bahwa bulan-bulan terjadinya intensitas *rain sum* yang tinggi ini adalah indikator dari musim hujan yang terjadi di kota tersebut. Selain itu, dibuat fitur kategori terkait dengan pasangan kota dan musim hujannya bernama *city_rainy_month*, sebagai contoh, jika pada tanggal tertentu Kota Su mengalami musim hujan, maka nilai kategorinya adalah su_rainy, jika sebaliknya maka nilai kategorinya su_not rainy. Selanjutnya pada data *train* atau *test* memiliki 2 fitur terkait waktu matahari terbit dan waktu matahari tenggelam, dengan menggunakan kedua informasi tersebut dibuat suatu fitur bernama *diff_sunrise_sunset* yang berisi selisih waktu (dalam detik) antara waktu matahari tenggelam dan waktu matahari terbit.

Untuk mengekstrak pola musiman yang tertangkap secara mingguan maupun bulanan, dibuat fitur baru terkait transformasi nilai dari minggu dan bulan ke fungsi sinus dan cosinus yang didefinisikan sebagai berikut

$$f(x) = \sin\left(\frac{x}{2 \times periode \times \pi}\right) \quad \textbf{(4.1.)}$$

$$g(x) = \cos\left(\frac{x}{2 \times periode \times \pi}\right) \quad \textbf{(4.2.)}$$

dengan periode adalah jumlah minggu atau bulan pada satu tahunnya yakni 52 atau 12 secara berturut-turut. Fitur-fitur ini dinamakan sebagai *week_sine*, *week_cos*, *month_sine,* dan *month_cos*.

V. PEMODELAN

5.1. Pembagian Data

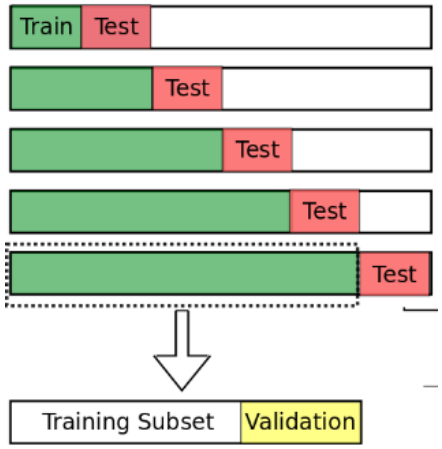
Pada tahap pemodelan, *train data* akan dibagi menjadi 2 subdata, yaitu data *train_subset* dan data *test_subset*. Data *train_subset* merupakan data historis dari tahun 2018 sampai 2020. Sementara, untuk data *test_subset* adalah data historis tahun 2021.

5.2. Model *Gradient Boosted Machine Learning*

Pemodelan dilakukan dengan melatih 2 model *gradient boosted machine learning* yaitu *Extreme Gradient Boosting* (XGBoost) dan *Light Gradient Boosting Machine* (LGBM). Untuk setiap model, dilakukan *hyperparameter tuning* untuk meminimumkan nilai MSE (*Mean Squared Error*).

5.3. *Hyperparameter Tuning*

Hyperparameter tuning adalah proses pencarian parameter yang dapat membuat model memiliki nilai metrik evaluasi terbaik. Pada penelitian ini, metode yang digunakan adalah *Randomized Search Cross Validation*, yaitu metode untuk mencari parameter terbaik dengan melakukan sampling pada kumpulan parameter yang dipilih untuk masing - masing fold *time series cross validation*.



Gambar V-1 Time series cross validation

Berikut adalah konfigurasi hyperparameter untuk tiap model,

Tabel V-1 Konfigurasi hyperparameter setiap model

No.	Parameter	Himpunan Nilai Parameter
1.	*Subsample*	{0.6, 0.7, 0.8, 0.9, 1.0}
2.	*n_estimators*	{900, 1100, 1300, 1500}
3.	*eta/learning_rate*	{0.01, 0.025, 0.05, 0.1}
4.	*max_depth*	{5, 7, 9, 11, 13, 15}
5.	*min_child_weight*	{1, 3, 5, 7, 9}
6.	*lambda*	{0.01, 0.03, 0.05, 0.07, 0.1, 1.0}
7.	*gamma*	{0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0}
8.	*colsample_bytree*	{0.6, 0.8, 1.0}
9.	*alpha*	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}

Tahap selanjutnya, dilakukan 100 iterasi untuk melakukan hyperparameter tuning, sehingga didapat nilai MSE terbaik dengan konfigurasi hyperparameter sebagai berikut.

Tabel V-2 Hasil hyperparameter tuning

No.	Parameter	Nilai Parameter	
		LGBM	XGBoost
1.	*subsample*	1	0.8
2.	*n_estimators*	2000	2000
3.	*eta/learning_rate*	0.025	0.025
4.	*max_depth*	13	5
5.	*min_child_weight*	1	3
6.	*lambda*	1	1
7.	*gamma*	-	1
8.	*colsample_bytree*	0.6	0.6
9.	*alpha*	0.4	0.1

Dengan konfigurasi *hyperparameter* di atas, diperoleh nilai MSE sekitar 13.61 untuk model LGBM dan untuk model XGBoost diperoleh nilai MSE sekitar 13.29.

VI. VALIDASI

6.1. Evaluasi pada Data *Time Series K-Fold*

Setelah dilakukan hyperparameter tuning pada seluruh data uji, dilakukan evaluasi model dengan mengkonfigurasi data train menjadi 3 *fold* dengan konfigurasi sebagai berikut.

- *Fold-1* dengan data latih pada tahun 2018 dan data uji pada tahun 2019
- *Fold-2* dengan data latih pada tahun 2018, 2019 dan data uji pada 2020

- *Fold-3* dengan data latih pada tahun 2018, 2019, 2020 dan data uji pada 2021

6.2. Hasil Evaluasi Model

Setelah dilakukan evaluasi dengan konfigurasi di atas didapat hasil untuk tiap fold nya sebagai berikut.

Tabel VI-1 Evaluasi K-Fold dengan Model LGBM dan XGBoost

No.	K-Fold	*Mean Squared Error* (MSE)	
		LGBM	XGBoost
1.	*Fold-1*	17.08	15.82
2.	*Fold-2*	23.08	22.24
3.	*Fold-3*	13.60	13.29

VII. PENUTUP

7.1. Kesimpulan

Berdasarkan pada Bab 6 Validasi, diperoleh performa model LGBM secara rata-rata MSE adalah 17.92 sementara pada model XGBoost diperoleh skor rata-rata MSE adalah 17.11.

LAMPIRAN

1. Link Repo GitHub: https://github.com/abilcode/weather-forecasting