

Analisis Perbandingan Model XGBoost dan Random Forest untuk Prediksi Potensi Banjir di DKI Jakarta

Alek Piter Wardoyo

Teknik Informatika Universitas Tarumanagara

alekpiterw1@gmail.com

ABSTRACT

Flooding in DKI Jakarta is a hydrometeorological disaster that requires an accurate early warning system. The main problem with current prediction systems is the limitation in integrating climate data from upstream (Bogor) with hydrological conditions downstream (Jakarta). This research aims to build and compare machine learning classification models to address this issue. The methods used are Extreme Gradient Boosting (XGBoost) and Random Forest, utilizing integrated data covering rainfall (Jakarta and Bogor) and Water Level (TMA) at the Katulampa and Manggarai floodgates for the period of January 2024 to March 2025. The models were evaluated using hyperparameter tuning (GridSearchCV, RandomizedSearchCV) on two data split schemes (70:30 and 80:20). The research results show that the XGBoost model with GridSearchCV on the 80/20 scheme provided the best performance with an accuracy of 82.61%. The best Random Forest model (GridSearchCV, 80/20) achieved an accuracy of 80.43%. Although XGBoost was numerically superior, the results of the McNemar's Test indicated that there was no statistically significant difference (calculated $\chi^2 0.25 < \text{table } \chi^2 3.841$) between the performance of the two models.

Keywords : *flood, warning system, rainfall, waterlevel.*

ABSTRAK

Banjir di DKI Jakarta merupakan bencana hidrometeorologis yang memerlukan sistem peringatan dini yang akurat. Permasalahan utama sistem prediksi saat ini adalah keterbatasan dalam mengintegrasikan data iklim di hulu (Bogor) dengan kondisi hidrologis di hilir (Jakarta). Penelitian ini bertujuan membangun dan membandingkan model klasifikasi machine learning untuk mengatasi masalah tersebut. Metode yang digunakan adalah Extreme Gradient Boosting (XGBoost) dan Random Forest, dengan memanfaatkan data terintegrasi mencakup curah hujan (Jakarta dan Bogor) serta Tinggi Muka Air (TMA) di Pintu Air Katulampa dan Manggarai periode Januari 2024 hingga Maret 2025. Model dievaluasi menggunakan hyperparameter tuning (GridSearchCV, RandomizedSearchCV) pada dua skema pembagian data (70:30 dan 80:20). Hasil penelitian menunjukkan bahwa model XGBoost dengan GridSearchCV pada skema 80/20 memberikan performa terbaik dengan akurasi 82,61%. Model Random Forest terbaik (GridSearchCV, 80/20) mencapai akurasi 80,43%. Meskipun XGBoost unggul secara numerik, hasil Uji McNemar menunjukkan bahwa tidak terdapat perbedaan yang signifikan secara statistik (χ^2 hitung 0,25 < χ^2 tabel 3,841) antara kinerja kedua model tersebut.

Kata kunci : *banjir, peringatan sistem, curah hujan, muka air.*

PENDAHULUAN

Banjir merupakan bencana hidrometeorologis yang paling sering melanda DKI Jakarta. Dampaknya signifikan terhadap aspek sosial, ekonomi, dan lingkungan. Faktor penyebab utamanya bersifat kompleks, mencakup curah hujan tinggi, limpasan air dari hulu, pengendapan sedimen, serta kondisi geografis Jakarta [1]. Kerentanan wilayah DKI Jakarta semakin diperburuk oleh kondisi geografisnya yang sekitar 40%

wilayahnya berada di bawah permukaan laut. Kondisi ini diperparah oleh masifnya alih fungsi lahan yang mengurangi daerah resapan air. Banjir besar pada awal tahun 2020, yang menyebabkan lebih dari 60 korban jiwa dan kerugian ekonomi diperkirakan mencapai Rp1,5 triliun, menegaskan urgensi penguatan sistem mitigasi dan peringatan dini[2].

Permasalahan utama dalam sistem peringatan dini saat ini adalah keterbatasan dalam memprediksi banjir secara akurat. Banyak sistem prediksi yang ada belum mampu mengintegrasikan faktor iklim di hulu (Bogor) dengan kondisi hidrologis di hilir (Jakarta). Padahal, intensitas hujan di Bogor secara langsung memengaruhi fluktuasi Tinggi Muka Air (TMA) Sungai Ciliwung. Pendekatan hidrologis konvensional seringkali kesulitan dalam memodelkan hubungan non-linear yang kompleks antara curah hujan, debit sungai, dan TMA.

Perkembangan teknologi *machine learning* menawarkan solusi untuk mengatasi kompleksitas ini. Algoritma *ensemble learning* seperti Extreme Gradient Boosting (XGBoost) dan Random Forest telah terbukti unggul dalam menangani data kompleks dengan akurasi tinggi. Beberapa penelitian sebelumnya telah menerapkan *machine learning* untuk prediksi banjir di Jakarta. Namun, masih terdapat kesenjangan penelitian (research gap), dimana sebagian besar studi terdahulu masih terbatas menggunakan satu jenis variabel saja, misalnya data cuaca saja, tanpa menggunakan data tinggi muka air di sepanjang aliran Sungai Ciliwung. Padahal, integrasi variabel hulu dan hilir sangat krusial untuk dinamika banjir Jakarta.

Berdasarkan latar belakang tersebut, penelitian ini mengisi kesenjangan penelitian dengan tujuan membangun dan membandingkan model klasifikasi menggunakan algoritma XGBoost dan Random Forest. Model ini dikembangkan dengan memanfaatkan kombinasi data curah hujan (Jakarta dan Bogor) serta data TMA (Pintu Air Katulampa dan Manggarai). Penelitian ini bertujuan mendapatkan model yang akurat, yang dievaluasi performanya menggunakan metrik Confusion Matrix, Accuracy, Precision, Recall, dan F1-Score

METODE PENELITIAN

Data yang digunakan adalah data sekunder dalam rentang waktu Januari 2024 hingga Maret 2025. Data iklim harian diperoleh dari situs data online resmi BMKG untuk Stasiun Meteorologi Kemayoran (Jakarta) dan Stasiun Meteorologi Citeko (Bogor) [3]. Data hidrologi (TMA) dan data catatan kejadian banjir diperoleh dari situs BPBD DKI Jakarta [4].

Tahap pra-pemrosesan data bertujuan untuk memastikan bahwa data yang digunakan memiliki kualitas yang baik, terstruktur, serta bebas dari kesalahan format maupun nilai-nilai yang tidak valid. Proses ini diawali dengan transformasi data, yang mencakup pembersihan struktur tabel, menyesuaikan nama kolom, serta mengonversi kolom numerik (seperti suhu dan curah hujan) dan kolom tanggal ke format datetime yang sesuai. Langkah selanjutnya adalah penanganan nilai hilang, di mana nilai anomali (seperti '8888') diubah menjadi NaN dan ditangani menggunakan metode interpolasi linear untuk menjaga kontinuitas dan tren data deret waktu. Terakhir, dilakukan integrasi dan pelabelan data; pada tahap ini, seluruh data iklim

dari Jakarta dan Bogor serta data TMA diintegrasikan berdasarkan kolom tanggal. Berdasarkan catatan kejadian banjir dari BPBD, dibuat variabel target (label) biner dengan kategori "Banjir" (nilai 1) dan "Tidak Banjir" (nilai 0). Dataset akhir yang dihasilkan terdiri dari 14 fitur input dan 1 variabel target yang siap digunakan untuk pemodelan. Dataset yang telah bersih dibagi menjadi data latih dan data uji menggunakan dua skema rasio: 70:30 dan 80:20. Daftar fitur yang digunakan sebagai variabel input (X) serta variabel target (y) dalam penelitian ini dapat dilihat pada Tabel 2

Tabel 2. Fitur Input (X) dan Target (y) Penelitian

Kategori Data	Nama Fitur	Deskripsi
Tinggi Muka Air	Bendung Katulampa	Bendung Katulampa (cm)
Tinggi Muka Air	Pos Depok	Pos Depok (cm)
Tinggi Muka Air	Manggarai BKB	Manggarai BKB (cm)
Tinggi Muka Air	PA. Karet	PA. Karet (cm)
Cuaca Bogor	TN_BGR	Suhu Min Bogor (°C)
Cuaca Bogor	TX_BGR	Suhu Max Bogor (°C)
Cuaca Bogor	TAVG_BGR	Suhu Rata-rata Bogor (°C)
Cuaca Bogor	RH_AVG_BGR	Kelembaban Rata-rata Bogor (%)
Cuaca Bogor	RR_BGR	Curah Hujan Bogor (mm)
Cuaca Jakarta	TN_JKT	Suhu Min Jakarta (°C)
Cuaca Jakarta	TX_JKT	Suhu Max Jakarta (°C)
Cuaca Jakarta	TAVG_JKT	Suhu Rata-rata Jakarta (°C)
Cuaca Jakarta	RH_AVG_JKT	Kelembaban Rata-rata Jakarta (%)
Cuaca Jakarta	RR_JKT	Curah Hujan Jakarta (mm)

1.1. XGBoost

Extreme Gradient Boosting (XGBoost) adalah algoritma ensemble berbasis gradient boosting yang dirancang untuk efisiensi, skalabilitas, dan performa tinggi [5]. XGBoost merupakan salah satu teknik pembelajaran mesin untuk mengatasi permasalahan regresi dan klasifikasi berdasarkan Gradient Boosting Decision Tree (GBDT) [6]. XGBoost bekerja dengan membangun model prediktif secara bertahap dalam bentuk deretan pohon keputusan (decision trees) yang dioptimalkan menggunakan metode additive training. Fungsi loss mengevaluasi perbedaan antara nilai sebenarnya dengan nilai hasil prediksi, sedangkan fungsi regularisasi menambahkan penalti terhadap kompleksitas pohon untuk menjaga model agar tidak terjadi overfitting. Fungsi objektif yang digunakan oleh algoritma XGBoost dapat dilihat pada Persamaan (2.1).

$$Obj(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i^t) + \sum_{k=1}^t \Omega(f_k) \quad (2.1)$$

Keterangan:

$l(y_i, \hat{y}_i^t)$ = fungsi loss
 \hat{y}_i^t = prediksi ke-t

f_k = pohon keputusan ke-k
 $\Omega(f_k)$ = fungsi regularisasi

Algoritma XGBoost menggunakan fungsi loss logistik untuk klasifikasi dua kelas. Turunan pertama dari fungsi loss logistik adalah gradient, Gradient menunjukkan arah dan perubahan yang harus dilakukan agar prediksi lebih mendekati target. Gradient didefinisikan dengan Persamaan (2.2).

$$g_i = \frac{\partial L}{\partial F} = p_i - y_i \quad (2.2)$$

Keterangan:

L = Fungsi loss (logistic loss dalam kasus klasifikasi biner).

F = Nilai prediksi logit

y_i = Label sebenarnya (0 atau 1).

p_i = Probabilitas hasil prediksi,

Turunan kedua dari fungsi loss logistik adalah hessian. Hessian menunjukkan kecepatan perubahan gradient atau “kelengkungan” fungsi loss. Hal ini berguna untuk menyesuaikan langkah pembaruan bobot supaya tidak terlalu besar/kecil, sehingga update jadi lebih stabil. Hessian didefinisikan dengan Persamaan (2.3).

$$h_i = \frac{\partial^2 L}{\partial F^2} = p_i(1 - p_i) \quad (2.3)$$

Keterangan:

L = Fungsi loss (logistic loss dalam kasus klasifikasi biner).

F = Nilai prediksi logit

y_i = Label sebenarnya (0 atau 1).

p_i = Probabilitas hasil prediksi,

Setiap pohon dalam XGBoost dibangun dengan menentukan bobot pada setiap daun (leaf). Bobot optimal pada sebuah leaf ditentukan berdasarkan nilai gradien dan hessian dari data yang masuk ke leaf tersebut. Rumus untuk menghitung bobot optimal dari setiap daun dapat dilihat pada Persamaan (2.4).

$$w_j^* = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (2.4)$$

Keterangan:

$\sum_{i \in I_j} g_i$ = Jumlah gradien dari semua sampel data

$\sum_{i \in I_j} h_i$ = Jumlah hessian dari semua sampel data

λ = Parameter regularisasi L2.

Untuk menentukan pemisahan (split) terbaik pada suatu node, digunakan metrik gain yang dihitung dari kontribusi total gradien dan hessian pada node kiri dan kanan. Gain dihitung menggunakan persamaan yang dapat dilihat pada Persamaan (2.5).

$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (2.5)$$

Keterangan:

G_L, H_L = total gradient dan hessian pada child kiri,

G_R, H_R = total gradient dan hessian pada child kanan.
 λ = Parameter regularisasi L2.
 γ = Biaya regularisasi

Split dengan nilai gain terbesar akan dipilih untuk membentuk struktur pohon yang paling informatif. Prediksi pada iterasi ke-t diperoleh dengan menambahkan kontribusi dari pohon baru terhadap hasil prediksi sebelumnya. Rumus prediksi pada iterasi ke-t dapat dilihat pada Persamaan (2.6).

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \quad (2.6)$$

Keterangan:

$\hat{y}_i^{(t)}$ = Prediksi model saat ini (pada iterasi ke-t)
 $\hat{y}_i^{(t-1)}$ = Prediksi model dari iterasi sebelumnya (pada iterasi t-1)
 η = Learning Rate
 $f_t(x_i)$ = Pohon keputusan baru (pohon ke-t)

Dengan mekanisme pembaruan ini, model XGBoost secara bertahap meningkatkan akurasi prediksi melalui proses boosting. Proses ini memastikan bahwa setiap pohon yang ditambahkan berfokus pada memperbaiki kesalahan dari iterasi sebelumnya sehingga model menjadi semakin kuat dan andal.

1.2. Random Forest

Random Forest merupakan kumpulan dari beberapa pohon keputusan (decision tree) yang digunakan untuk membuat prediksi dengan memecahkan data menjadi beberapa kategori berdasarkan atribut tertentu dan membuat keputusan berdasarkan perbandingan nilai tertentu [7]. Algoritma ini bekerja dengan membangun sejumlah besar pohon keputusan secara acak dan kemudian menggabungkan hasil prediksinya untuk meningkatkan akurasi serta mengurangi risiko overfitting [8].

Secara matematis, proses pembangunan pohon keputusan pada Random Forest menggunakan konsep entropy untuk mengukur tingkat ketidakpastian suatu himpunan data. Rumus entropy dapat dilihat pada Persamaan (2.7).

$$\text{Entropy}(S) = \sum_{i=1}^c -p_i \log_2 p_i \quad (2.7)$$

Keterangan:

S = himpunan dataset
 C = jumlah kelas
 p_i = probabilitas frekuensi kelas ke-i dalam dataset

Selanjutnya, pemilihan atribut terbaik pada pembentukan pohon keputusan ditentukan melalui nilai information gain. Rumus information gain dapat dilihat pada Persamaan (2.8).

$$\text{Gain} = \text{Entropy}(S) - \sum_{i=1}^k \frac{|S_i|}{|S|} \times \text{Entropy}(S_i) \quad (2.8)$$

Keterangan:

- S = himpunan dataset
- $|S_i|$ = jumlah sampel untuk nilai i
- $|S|$ = jumlah seluruh sampel dalam dataset
- $Entropy(S_i)$ = nilai entropy untuk sampel yang memiliki nilai i

Untuk kasus klasifikasi, prediksi akhir dari Random Forest diambil dari hasil voting mayoritas seluruh pohon. Hasil voting untuk klasifikasi dengan Random Forest dirumuskan dengan Persamaan (2.9).

$$\hat{y} = mode\{T_1(x), T_2(x), \dots, T_B(x)\} \tag{2.9}$$

Keterangan:

- \hat{y} = himpunan dataset
- $T_i(x)$ = jumlah sampel untuk nilai i
- B = jumlah seluruh sampel dalam dataset
- $mode\{.\}$ = nilai entropy untuk sampel yang memiliki nilai i

Kekuatan dari Random Forest terletak pada kemampuan setiap pohon yang dibangun secara independen, serta korelasi antar pohon yang berbeda, sehingga menghasilkan model klasifikasi yang lebih akurat dan stabil. Selain itu, Random Forest juga mampu menangani dataset berukuran besar dengan variabel yang beragam, mengurangi risiko overfitting dibandingkan dengan pohon keputusan tunggal, serta memberikan estimasi tingkat kepentingan setiap variabel dalam proses prediksi. Kemampuan ini menjadikan Random Forest sebagai salah satu algoritma yang handal untuk permasalahan klasifikasi maupun regresi, khususnya pada data yang bersifat kompleks dan memiliki ketidakseimbangan kelas.

1.3. Metode Evaluasi

Confusion matrix adalah alat evaluasi kinerja model klasifikasi yang menyajikan perbandingan antara prediksi model dan nilai aktual. Contoh confusion matrix untuk klasifikasi untuk dua kelas, yaitu kelas 0 dan kelas 1 dapat dilihat pada Tabel 2.2 [9].

Tabel 2.2 Confusion Matrix

		Kelas Prediksi	
		0	1
Kelas Sebenarnya	0	TP	FN
	1	FP	TN

Keterangan:

- TP (True Positive) = jumlah data dari kelas 1 yang benar diklasifikasikan sebagai kelas 1
- TN (True Negative) = jumlah data dari kelas 0 yang benar diklasifikasikan sebagai kelas 0
- FP (False Positive) = jumlah data dari kelas 0 yang salah diklasifikasikan sebagai kelas 1
- FN (False Negative) = jumlah data dari kelas 1 yang salah diklasifikasikan sebagai kelas 0

Confusion Matrix juga memiliki beberapa metrik turunan yang digunakan untuk mengevaluasi kinerja model klasifikasi, yaitu accuracy, precision, recall dan F-1 score. Accuracy menunjukkan tingkat ketepatan model dalam mengklasifikasikan data dengan benar. Accuracy dapat didefinisikan dengan Persamaan (2.7)

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (2.10)$$

Precision menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Precision dapat didefinisikan dengan Persamaan (2.8)

$$Precision = \frac{TP}{(TP + FP)} \quad (2.11)$$

Recall mengukur keberhasilan model dalam menemukan kembali sebuah informasi. Recall dapat didefinisikan dengan Persamaan (2.9)

$$Recall = \frac{TP}{(TP + FN)} \quad (2.12)$$

F-1 Score merupakan nilai rata-rata precision dan recall. F-1 Score dapat didefinisikan dengan persamaan (2.10)

$$F - 1 \text{ Score} = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (2.13)$$

Nilai F1-Score berada pada rentang 0 hingga 1, dengan nilai terbaik sebesar 1 yang berarti kinerja model sangat baik dan nilai terburuk sebesar 0 yang berarti kinerja model sangat buruk.

1.4. Uji McNemar

Uji McNemar adalah uji statistik non-parametrik yang digunakan secara khusus untuk menguji perbedaan signifikan antara dua model klasifikasi pada data berpasangan (paired data). Penggunaan data berpasangan berarti kedua model (misalnya, Model A dan Model B) diuji pada dataset yang sama persis (Dietterich, 1998). Uji ini berakar pada teori statistical hypothesis testing dan dikembangkan oleh McNemar (1947).

Secara operasional, Uji McNemar berfokus pada ketidaksetujuan (*disagreements*) antara kedua model. Hasil prediksi kedua model pada data uji diringkas dalam tabel kontingensi 2x2 pada Tabel 2.3.

Tabel 2.3. Tabel Kontingensi 2x2

	Model B: Benar	Model B: Salah
Model A: Benar	a	b
Model A: Salah	c	d

Keterangan:

a = Jumlah data di mana Model A dan Model B sama-sama memprediksi dengan benar.

b = Jumlah data di mana Model A benar, tetapi Model B salah.

c = Jumlah data di mana Model A salah, tetapi Model B benar.

d = Jumlah data di mana Model A dan Model B sama-sama memprediksi dengan salah.

Uji McNemar mengevaluasi hipotesis nol bahwa kedua model memiliki tingkat kesalahan (error rate) yang sama [29]. Hipotesis nol mengasumsikan bahwa jumlah kesalahan yang dibuat satu model tetapi tidak dibuat oleh model lainnya adalah sama. Hipotesis alternatifnya adalah tingkat kesalahan keduanya berbeda. Uji ini mengabaikan kasus 'a' dan 'd' (di mana kedua model setuju), karena kasus tersebut tidak memberikan informasi tentang perbedaan kinerja di antara keduanya. Statistik uji McNemar dihitung hanya berdasarkan kasus ketidaksetujuan 'b' dan 'c' dengan distribusi Chi-Square dan 1 derajat kebebasan (degree of freedom).

Formula dasar McNemar dirumuskan seperti pada persamaan (2.14)

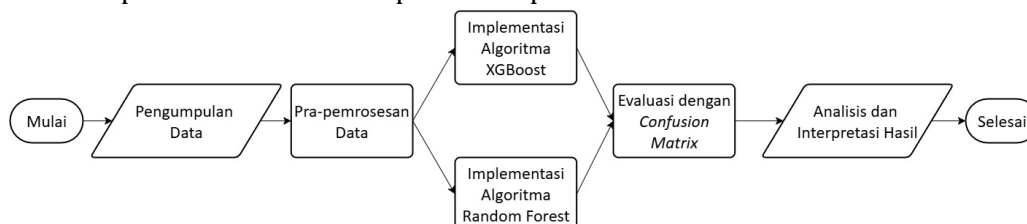
$$x^2 = \frac{(|b - c| - 1)^2}{b + c} \tag{2.14}$$

dengan b adalah jumlah instance di mana model A benar dan model B salah, dan c adalah jumlah instance di mana model A salah dan model B benar. Jika nilai x^2 yang dihitung melebihi nilai kritis (misalnya, nilai x^2 tabel = 3.84 untuk tingkat signifikansi $\alpha = 0.05$), maka H_0 ditolak, yang mengindikasikan adanya perbedaan kinerja yang signifikan secara statistik antara kedua model.

Uji McNemar memberikan interpretasi yang lebih adil dan reliabel untuk perbandingan dua model pada dataset yang sama dibandingkan uji t-student berpasangan pada akurasi, karena uji ini memperhitungkan korelasi antar hasil prediksi dan fokus pada perbedaan kesalahan klasifikasi secara langsung [10].

1.5. Skema Penelitian

Skema penelitian dalam penelitian ini disusun untuk memberikan gambaran sistematis mengenai alur berpikir penelitian dalam merumuskan solusi atas permasalahan yang diteliti. Penelitian ini diawali dengan pengumpulan data curah hujan dan tinggi muka air (TMA) yang relevan dengan kejadian banjir. Selanjutnya dilakukan pra-pemrosesan data untuk memastikan kualitas data yang digunakan dalam pemodelan. Tahap berikutnya adalah implementasi algoritma Random Forest dan XGBoost sebagai model prediksi banjir. Hasil prediksi dari kedua model tersebut kemudian dievaluasi menggunakan confusion matrix untuk mengukur performa melalui indikator akurasi, precision, recall, dan F1-score. Tahap terakhir adalah analisis dan interpretasi hasil untuk membandingkan kinerja kedua algoritma sehingga diperoleh kesimpulan yang tepat. Gambaran skema penelitian tersebut dapat dilihat pada Gambar 2.1.



Gambar 2.1 Diagram Skema Penelitian

HASIL DAN PEMBAHASAN

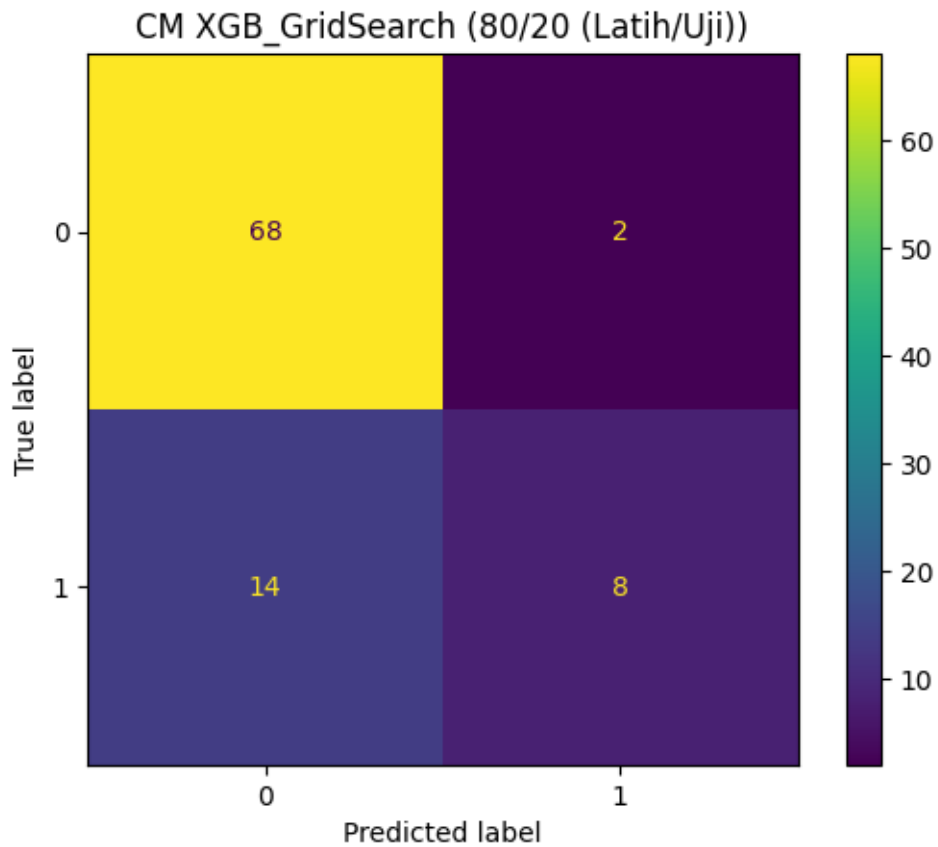
Proses pra-pemrosesan data berhasil menghasilkan dataset terintegrasi yang terdiri dari 456 entri data harian (Jan 2024 - Mar 2025) dengan 14 fitur input dan 1 target. Total 12 skenario eksperimen telah dilakukan, mengombinasikan 2 algoritma (XGBoost, Random Forest), 3 metode tuning (Default, GridSearchCV, RandomizedSearchCV), dan 2 skema data (70:30, 80:20).

Hasil perbandingan keseluruhan menunjukkan bahwa skenario pembagian data 80/20 secara konsisten memberikan hasil yang lebih baik daripada skema 70/30, baik untuk XGBoost maupun Random Forest. Metode tuning GridSearchCV juga terbukti menghasilkan akurasi tertinggi untuk kedua algoritma

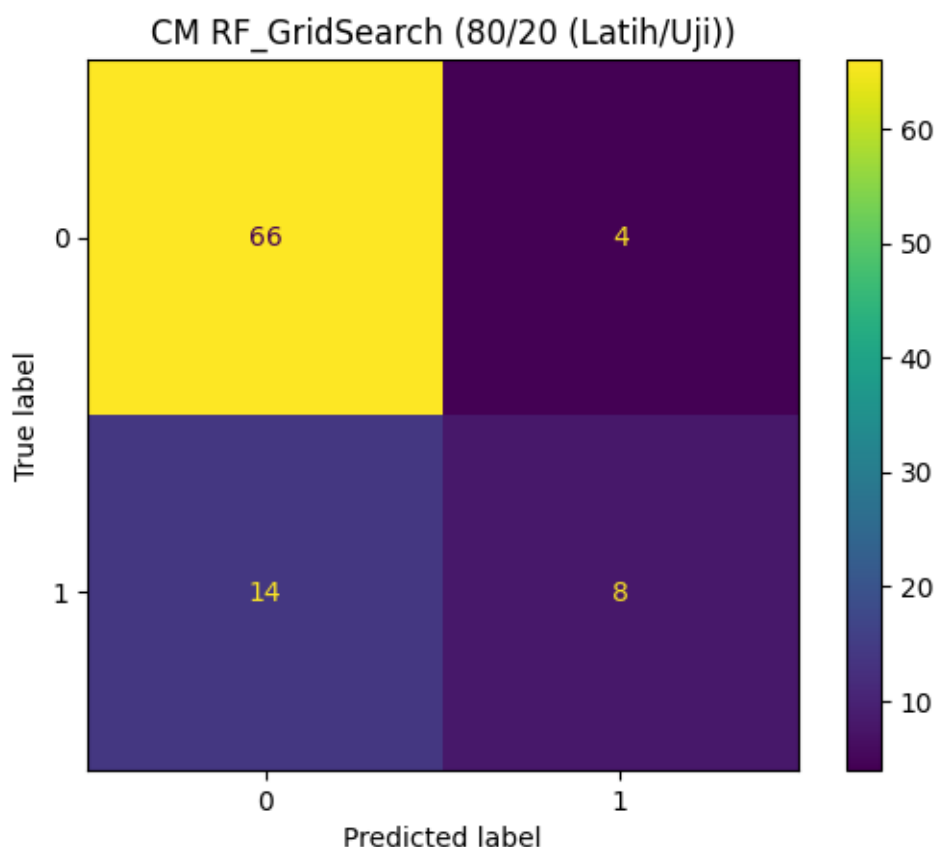
Peringkat	Algoritma & Tuning	Skenario Data	Akurasi	
1	XGB_GridSearch	80/20 (Latih/Uji)	0,826087	19,92
2	XGB_GridSearch	70/30 (Latih/Uji)	0,824818	23,31
3	RF_RandomSearch	80/20 (Latih/Uji)	0,804348	92,90
4	RF_GridSearch	80/20 (Latih/Uji)	0,804348	158,71
5	RF_RandomSearch	70/30 (Latih/Uji)	0,802920	89,94
6	RF_GridSearch	70/30 (Latih/Uji)	0,802920	140,33
7	RF_Default	80/20 (Latih/Uji)	0,793478	0,23
8	XGB_Default	80/20 (Latih/Uji)	0,782609	0,11
9	XGB_RandomSearch	80/20 (Latih/Uji)	0,782609	15,73
10	RF_Default	70/30 (Latih/Uji)	0,781022	0,2
11	XGB_RandomSearch	70/30 (Latih/Uji)	0,773723	23,31
12	XGB_Default	70/30 (Latih/Uji)	0,773723	0,06

Model XGBoost terbaik dicapai menggunakan GridSearchCV pada skema data 80/20, menghasilkan akurasi tertinggi sebesar 82,61% dengan parameter optimal $learning_rate = 0.05$, $max_depth = 3$, $n_estimators = 50$, dan $subsample = 0.8$. Serupa dengan itu, model Random Forest terbaik juga dicapai menggunakan GridSearchCV pada skema data 80/20, dengan akurasi 80,43%. Dari sisi efisiensi, XGBoost menunjukkan keunggulan signifikan; waktu pelatihan untuk model XGBoost terbaik

adalah 19,92 detik, jauh lebih cepat dibandingkan model Random Forest terbaik yang membutuhkan waktu 158,71 detik.



Visualisasi *confusion matrix* dari model XGBoost terbaik menunjukkan model mampu memprediksi 68 kasus 'Tidak Banjir' (TN) dan 8 kasus 'Banjir' (TP) dengan benar. Namun, model ini masih melakukan 14 kesalahan 'False Negative' (FN) memprediksi 'Tidak Banjir' padahal sebenarnya 'Banjir' dan 2 kesalahan 'False Positive' (FP)



Model Random Forest terbaik menunjukkan 66 TN dan 8 TP. Model ini memiliki kesalahan FN yang sama (14 kasus) namun kesalahan FP yang sedikit lebih tinggi (4 kasus). Secara numerik, XGBoost (82,61%) lebih unggul daripada Random Forest (80,43%).

Namun, untuk menentukan apakah perbedaan ini signifikan secara statistik, dilakukan Uji McNemar. Uji McNemar dilakukan dengan membandingkan prediksi dari dua model terbaik tersebut pada 92 data uji (skema 80/20). Tabel kontingensi 2x2 merangkum hasil perbandingan

	Random Forest Benar	Random Forest Salah
XGBoost Benar	73	3
Xgboost Salah	1	15

Statistik uji McNemar dihitung berdasarkan kasus ketidaksetujuan (b dan c):

$$x^2 = \frac{(|b-c|-1)^2}{b+c} = \frac{(|3-1|-1)^2}{3+1} = \frac{1^2}{4} = 0,25$$

Nilai x^2 hitung adalah 0,25. Dengan tingkat signifikansi (α) = 0,05 dan derajat kebebasan (df) = 1, nilai x^2 tabel (kritis) adalah 3,841. Karena x^2 hitung (0,25) < x^2 tabel (3,841), maka H_0 diterima. Hal ini menunjukkan bahwa tidak terdapat perbedaan yang signifikan secara statistik antara performa model XGBoost (akurasi 82,61%) dan model Random Forest (akurasi 80,43%) dalam memprediksi kejadian banjir menggunakan dataset ini.

KESIMPULAN

Berdasarkan hasil penelitian dan analisis data, dapat ditarik kesimpulan bahwa penelitian ini berhasil membangun model klasifikasi untuk prediksi potensi banjir harian di DAS Ciliwung hilir. Model ini sukses mengintegrasikan data Tinggi Muka Air (TMA) dari hulu (Katulampa, Depok) dan hilir (Manggarai, Karet), serta data cuaca dari Bogor (Citeko) dan Jakarta (Kemayoran). Berdasarkan 12 skenario pengujian, model XGBoost yang dioptimalkan menggunakan GridSearchCV dengan skema pembagian data 80% latih dan 20% uji menghasilkan performa terbaik dan mencapai akurasi 82,61%. Model Random Forest terbaik (juga dengan GridSearchCV dan skema 80/20) mencapai akurasi 80,43%. Hasil pengujian hipotesis menggunakan Uji McNemar untuk membandingkan kedua model terbaik tersebut menunjukkan nilai χ^2 hitung (0,25) lebih kecil dari nilai χ^2 tabel (3,841). Dengan demikian, H_0 diterima, yang berarti tidak terdapat perbedaan yang signifikan secara statistik antara performa model XGBoost dan Random Forest dalam kasus ini.

DAFTAR PUSTAKA

- Badan Meteorologi, Klimatologi, dan Geofisika (BMKG), "Situs data online BMKG" [Online]. Available: <https://dataonline.bmkg.go.id>.
- Badan Penanggulangan Bencana Daerah (BPBD) DKI Jakarta, "Water level monitoring" [Online]. Available: <https://bpbd.jakarta.go.id/waterlevel>.
- D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes classifier dan confusion matrix pada analisis sentimen berbasis teks pada Twitter," J-SAKTI (J. Sains Komput. Inform.), vol. 5, no. 2, pp. 697–711, 2021.
- I. L. Cherif and A. Kortebi, "On using eXtreme Gradient Boosting (XGBoost) machine learning algorithm for home network traffic classification," in 2019 Wireless Days (WD), 2019, pp. 1–6.
- L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- M. D. Arvi, R. M. Y. Sibarani, Y. I. Tanjung, and T. Fairuz, "Analisis faktor penyebab bencana banjir di kota-kota besar Indonesia: Studi kasus analisis banjir berbasis literasi," *Indones. J. Emerg. Trends Community Empower.*, vol. 3, no. 1, pp. 1–8, 2025.
- NPB, "Dampak Banjir Jakarta 2020: Kerugian Ekonomi dan Korban Jiwa," *Laporan Resmi BNPB*, Jan. 2020. [Online]. Available: <https://bnpb.go.id>
- P. Dangeti, *Statistics for Machine Learning: Build Supervised, Unsupervised, and Reinforcement Learning Models Using Both Python and R*. Birmingham, U.K.: Packt Publishing Ltd., 2017.
- T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 785–794.
- V. García, J. Luengo, and F. Herrera, "Advanced Statistical Tests for Comparing Machine Learning Models," *Information Sciences*, vol. 589, pp. 72–90, 2022.