

## Flood Status Prediction Based on Water Level Data Using Machine Learning Models

Aisyah Putri Widyastuti <sup>1\*</sup>, Sindhu Rakasiwi <sup>2\*\*</sup>

\* Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang  
[111202214169@mhs.dinus.ac.id](mailto:111202214169@mhs.dinus.ac.id)<sup>1</sup>, [sindhu.rakasiwi@dsn.dinus.ac.id](mailto:sindhu.rakasiwi@dsn.dinus.ac.id)<sup>2</sup>

### Article Info

#### Article history:

Received 2026-04-20  
Revised 2026-06-16  
Accepted 2026-06-18

#### Keyword:

*Adasyn,*  
*Flood,*  
*LSTM,*  
*Machine Learning,*  
*Prediction.*

### ABSTRACT

Flooding is one of the hydrometeorological disasters that frequently occurs in Indonesia and causes various social and economic losses. This study aims to compare the performance of five machine learning algorithms, namely Random Forest, Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Logistic Regression, as well as one Long Short-Term Memory (LSTM) deep learning model in predicting flood status based on water level data from seven observation posts in the DKI Jakarta area and its surroundings. The research stages include data preprocessing, handling unbalanced data using ADASYN, hyperparameter tuning, and evaluation using accuracy, precision, recall, and F1-score. To avoid data leakage, the data division process is carried out before preprocessing and oversampling. The results show that XGBoost produces the best performance with 96.0% accuracy, 95.5% precision, 96.9% recall, and 96.2% F1-score after hyperparameter tuning. The LSTM model also demonstrated competitive performance with an accuracy of 94.5% and an F1-score of 94.5%. Learning curve analysis showed that all models exhibited normal learning patterns with no indication of data leakage. The results indicate that XGBoost and LSTM have good potential for application in flood early warning systems based on water level data.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

### I. PENDAHULUAN

Banjir merupakan salah satu bencana hidrometeorologi yang paling sering terjadi di berbagai wilayah dunia, khususnya di negara beriklim tropis seperti Indonesia. Banjir salah satu bencana yang menjadi masalah bagi sebagian masyarakat yang tinggal di daerah dataran rendah maupun dibantaran sungai[1]. Untuk daerah perkotaan yang padat terutama Jakarta seringkali mengalami bencana banjir[2]. Peningkatan intensitas curah hujan, perubahan tata guna lahan, urbanisasi yang pesat, serta degradasi sistem drainase menyebabkan frekuensi dan dampak banjir terus meningkat setiap tahunnya[3]. Berdasarkan sebab dampak banjir yang meningkat menyebabkan ketinggian air yang semakin meningkat. Dampak yang ditimbulkan tidak hanya berupa kerugian ekonomi dan kerusakan infrastruktur, tetapi juga ancaman terhadap keselamatan jiwa dan stabilitas sosial masyarakat. Dalam Sistem mitigasi modern, ketinggian muka air sungai atau pintu air menjadi indikator operasional utama

dalam menentukan status waspada terhadap banjir karena mampu mempresentasikan kondisi hidrologi secara langsung[4]. Namun demikian, proses penentuan status banjir berdasarkan data ketinggian air masih menghadapi tantangan, seperti ketidakpastian data hidrologi, kompleksitas hubungan non-linear antar variabel lingkungan, serta keterbatasan metode konvensional dalam memodelkan pola data yang kompleks[5].

Perkembangan teknologi komputasi dan *machine learning* telah memberikan kontribusi yang cukup signifikan[6], terutama dalam peningkatan akurasi prediksi banjir[7]. Berbagai metode telah digunakan sesuai perkembangan diantaranya berbasis *Artificial Neural Network* (ANN) dan *Bayesian Neural Network* (BNN) yang mampu memodelkan ketidakpastian data hidrologi secara adaptif dibandingkan metode statistik konvensional[3]. Selain itu, pendekatan berbasis *Random Forest* [6], *Support Vector Machine*[8], dan *K-Nearest Neighbour* yang telah digunakan dalam prediksi banjir dan tipe banjir karena kemampuannya menangani

variabel multivariat[9]. Pendekatan klasifikasi probalistik seperti *Gaussian Naïve Bayes* juga digunakan untuk klasifikasi banjir jangka pendek meskipun performanya sangat bergantung pada jumlah data latih[10].

Beberapa penelitian juga membandingkan berbagai algoritma untuk meningkatkan performa prediksi banjir. *Support Vector Machine* (SVM) dilaporkan memiliki akurasi lebih tinggi dibandingkan *K-Nearest Neighbour* dalam prediksi curah hujan berpotensi banjir[9]. Namun, didalam penelitian lain juga menunjukkan bahwa *K-Nearest Neighbour* memiliki hasil yang lebih baik dari pada *Naïve Bayes*[11]. Perbandingan model *Naïve Bayes*, *K-Nearest Neighbour* dan *Random Forest* juga dilakukan oleh *Sadam* yang menunjukkan *Random Forest* model terbaik disusul *K-Nearest Neighbour* yang menunjukkan performa yang baik juga dibandingkan *Naïve Bayes*[12]. Penelitian tentang banjir yang berada di Desa Danyehkolot juga membandingkan model *Random Forest*, *Decision Tree*, dan *Naïve Bayes* yang menghasilkan bahwa model *Random Forest* merupakan model terbaik[13]. Menurut penelitian, menyatakan bahwa model *Random Forest* merupakan model yang terbaik daripada *K-Means* dalam memprediksi daerah rawan banjir yang berada di Jakarta[14]. Penelitian, menggunakan model *Gradient Boosting* dan *Random Forest* untuk memprediksi daerah rawan banjir yang berada di Kota Surabaya menggunakan data spasial. Perbandingan model juga dilakukan pada penelitian, yang menyatakan bahwa *Random Forest* merupakan model yang lebih baik daripada *Naïve Bayes*[15]. Namun model *XGBoost* juga digunakan dalam memprediksi banjir yang ada di Bandar Lampung yang menghasilkan model yang baik pula daripada *Arima* dan *Random Forest* [16]. Selain beberapa model diatas, penelitian lain menggunakan *Logistic Regression* merupakan model yang baik juga dalam memprediksi daerah banjir[17].

Seiring perkembangan deep learning, metode Long Short-Term Memory (LSTM) juga mulai banyak diterapkan dalam prediksi banjir dan curah hujan karena kemampuannya dalam menangkap pola data time series dan dependensi jangka panjang. Penelitian oleh Ahmad Wijayanto dkk. menggunakan algoritma LSTM dan Isolation Forest untuk identifikasi dini curah hujan berpotensi banjir di wilayah Semarang dan menunjukkan bahwa LSTM mampu menghasilkan prediksi curah hujan jangka panjang dengan nilai evaluasi yang baik[18]. Selain itu, penelitian Hadi Kardhana dkk. memanfaatkan model LSTM berbantuan HEC-HMS untuk meningkatkan prediksi banjir Jakarta dan menunjukkan kemampuan model dalam melakukan forecasting hingga enam jam ke depan[19]. Penelitian lain juga menunjukkan bahwa LSTM memiliki performa lebih baik dibandingkan Recurrent Neural Network (RNN) dalam prediksi curah hujan karena lebih stabil dalam mempelajari pola perubahan cuaca dan data deret waktu[20].

Berdasarkan gap tersebut, penelitian ini bertujuan untuk mengembangkan sistem prediksi status banjir berbasis machine learning dan deep learning dengan memanfaatkan data ketinggian air dari tujuh pos pengamatan di wilayah DKI

Jakarta dan sekitarnya. Penggunaan data dari multiple stations memungkinkan model untuk menangkap pola spasial kejadian banjir dan interaksi antar pos pengamatan yang berbeda. Untuk pemodelan, penelitian ini mengimplementasikan lima algoritma machine learning yang mewakili berbagai paradigma pembelajaran, yaitu Logistic Regression sebagai model linear yang sederhana dan mudah diinterpretasikan, K-Nearest Neighbors yang melakukan klasifikasi berdasarkan kemiripan data, Support Vector Machine yang efektif dalam membentuk batas keputusan optimal, Random Forest yang memanfaatkan pendekatan ensemble untuk meningkatkan stabilitas prediksi, serta XGBoost yang mampu menangani pola data kompleks melalui teknik gradient boosting. Selain itu, penelitian ini juga menerapkan metode Long Short-Term Memory (LSTM) sebagai representasi pendekatan deep learning yang memiliki kemampuan dalam mempelajari hubungan non-linear dan pola temporal pada data hidrologi, sehingga dapat digunakan sebagai pembanding terhadap model machine learning konvensional.

Oleh karena itu, penelitian ini bertujuan untuk menganalisis dan membandingkan performa kelima model machine learning, yaitu *Random Forest*, *Support Vector Machine* (SVM), *K-Nearest Neighbour* (KNN), *Logistic Regression*, *Extreme Gradient Boosting* (XGBoost), dan *Long Short-Term Memory* (LSTM) dalam memprediksi status. Evaluasi model dilakukan dengan menggunakan matriks performa seperti akurasi, presisi, *recall*, dan *F1-score*, yang secara menyeluruh memberikan gambaran yang lebih menyeluruh mengenai ketepatan dan stabilitas prediksi yang dihasilkan oleh masing-masing algoritma. Dengan demikian, penelitian ini diharapkan untuk dapat memberikan kontribusi signifikan terhadap pengembangan sistem peringatan dini terhadap bencana banjir berbasis *machine learning* yang lebih akurat dan adaptif, tetapi juga lebih aplikatif dalam mendukung mitigasi risiko bencana, khususnya wilayah yang rentan terhadap banjir.

## II. METODE

Penelitian ini bertujuan untuk mengembangkan sistem prediksi status banjir menggunakan metode *machine learning* pada dataset yang berisi data ketinggian air dari berbagai titik pemantauan di daerah rawan banjir. Penelitian ini dilakukan dengan serangkaian tahapan yang sistemasi, mulai dari pengumpulan data, eksplorasi data (EDA), *preprocessing*, hingga evaluasi model yang disajikan pada Gambar 1.

### A. Pengumpulan Dataset

Tahap pertama dalam penelitian ini adalah pengumpulan dataset. Tahapan pengumpulan dataset ini merupakan tahapan terpenting dalam proses pengolahan *machine learning*. Dalam penelitian ini dataset publik dari website kaggle yang digunakan berisi informasi mengenai ketinggian air yang diukur pada beberapa lokasi rawan banjir di Jakarta, seperti Katulampa, Pos Depok, Manggarai, Istiqlal, Jembatan Merah, Flusing Ancol, dan Marina Ancol. Dataset ini disimpan dalam

format CSV yang berisi rekaman berkala ketinggian air beserta label 'Status Banjir' sebagai variabel target.



Gambar 1. Tahapan Pengolahan Data

Setiap baris pada dataset mempresentasikan satu rekaman pengukuran yang muat tanggal, waktu, nilai ketinggian air masing – masing pos pemantauan, serta label label target biner Status Banjir (0 = tidak banjir, 1 = banjir ). Data ini bersifat time – serie dengan interval pencatatan yang reguler sehingga mencerminkan dinamika ketinggian air secara temporal dan spasial di sepanjang aliran sungai.

### B. EDA

Tahap Exploratory Data Analysis (EDA) dilakukan untuk memahami karakteristik dataset sebelum dilakukan pemodelan. Pada tahap ini, beberapa analisis dilakukan secara sistematis menggunakan library pandas, matplotlib, dan seaborn. Rangkaian EDA yang dilakukan mencakup :

- Cek informasi umum dataset menggunakan `df.info()` untuk mengetahui tipe data, jumlah kolom, dan keberadaan missing value pada setiap fitur.
- Analisis statistik deskriptif menggunakan `df.describe()` untuk memperoleh gambaran distribusi nilai minimum, maksimum, rata – rata, dan standar deviasi pada masing – masing fitur.
- Visualisasi distribusi data melalui histogram untuk setiap fitur ketinggian air untuk mengetahui pola persebaran dan *skewness* (kemencengan) data.
- Analisis tren ketinggian air terhadap waktu menggunakan line plot untuk mengamati pola temporal dari seluruh pos pemantauan secara bersamaan.
- Deteksi outlier menggunakan bloxplot untuk mengidentifikasi nilai – nilai ekstrem yang berpotensi mengganggu performa model.

- Analisis korelasi antar fitur menggunakan heatmap matriks korelasi untuk mengidentifikasi hubungan linear antara ketinggian air di setiap pos dan korelasinya dengan Status Banjir.
- Visualisasi hubungan antara ketinggian air masing – masing pos dengan Status Banjir menggunakan boxplot per kelas untuk memahami fitur yang paling diskriminatif.

Hasil EDA memberikan pemahaman menyeluruh tentang pola data dan menjadi dasar pengambilan keputusan dalam tahap berikutnya.

### C. Preprocessing

Setelah eksplorasi data, tahapan berikutnya yaitu preprocessing, merupakan serangkaian proses untuk mempersiapkan data agar bersih, konsisten, dan siap digunakan dalam pembuatan model. Pada tahap ini, beberapa langkah penting dilakukan :

#### a. Penanganan Missing Value

Missing value merupakan nilai yang kosong pada beberapa kolom yang kemudian dapat diatasi dengan menggunakan beberapa cara, salah satunya dengan menggunakan nilai median. Penanganan dengan nilai median itu sendiri merupakan penanganan yang menggunakan nilai tengah dari suatu variabel.

#### b. Penanganan Nilai Negatif

Kolom Katulampa ditemukan memiliki nilai negatif yang secara fisik tidak mungkin terjadi karena ketinggian air tidak dapat bernilai negatif. Nilai – nilai tersebut diganti menjadi 0 menggunakan fungsi `clip(lower=0)` untuk memastikan konsistensi data dengan kondisi nyata.

#### c. Penanganan Outlier

Outlier ekstrem ditangani menggunakan metode IQR Capping (*Interquartile Range*) dengan faktor 1,5. Pada metode ini, nilai yang berada di bawah batas bawah pada rumus (1) atau di atas batas atas pada rumus (2) diubah menjadi nilai batas tersebut (capping). Metode ini diterapkan pada kolom Manggarai, Jembatan Merah, Marina Ancol, Pos Depok, Flusing Ancol, dan Katulampa.

#### d. Transformasi Box-Cox

Transformasi Box-Cox diterapkan pada fitur Manggarai, Katulampa, dan Jembatan Merah untuk mengurangi *skewness* (kemencengan distribusi) dan membuat distribusi data lebih mendekati distribusi normal. Nilai +1 ditambahkan sebelum transformasi untuk menghindari dari nilai nol. Transformasi ini penting untuk meningkatkan performa beberapa algoritma *machine learning* yang sensitif terhadap distribusi data sehingga bekerja lebih optimal.

#### e. Normalisasi Fitur

Seluruh fitur numerik dinormalisasi menggunakan `StandardScaler` dari library `scikit-learn`. Proses ini mengubah

setiap fitur sehingga memiliki rata – rata 0 dan standar deviasi 1, yang penting untuk memastikan bahwa skala antar fitur tidak mempengaruhi bobot model secara tidak proporsional.

#### f. Ekstraksi Fitur Temporal

Kolom Tanggal dan Waktu digabungkan dan dikonversi menjadi objek datetime, kemudian diekstraksi menjadi fitur temporal baru, yaitu Hour (jam), Day (hari), dan Month (bulan). Fitur – fitur ini diharapkan dapat menangkap pola periodik kejadian banjir. Setelah diekstraksi, kolom Tanggal dan Waktu dihapus dari dataset.

#### g. Imbalance Dataset

Hasil EDA menunjukkan adanya ketidakseimbangan distribusi kelas pada variabel target Status Banjir, dimana jumlah sampel kelas tidak banjir lebih banyak dibandingkan kelas banjir. Kondisi ini dapat menyebabkan model cenderung memprediksi kelas mayoritas dan mengabaikan kelas minoritas.

Untuk mengatasi masalah tersebut, digunakan teknik *Adasyn* dari library *imbalanced-learn*. *Adasyn* bekerja dengan membuat sampel sintetis baru pada kelas minoritas berdasarkan interpolasi antara sampel yang ada dan tetangga terdekatnya di ruang fitur. *Adasyn* hanya diterapkan pada data latin (*train*) untuk menghindari kebocoran informasi pada data uji.

Pembagian data dilakukan sebelum penerapan *Adasyn* dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian menggunakan *train\_test\_split* dengan *random\_state=42*. *Adasyn* kemudian diterapkan pada *X\_train* dan *y\_train* untuk menghasilkan pelatihan data yang seimbang.

#### D. Pemodelan dan Pelatihan

Pemodelan dan pelatihan dalam penelitian ini dilakukan dengan menggunakan lima algoritma *machine learning* yang mewakili berbagai pendekatan dalam klasifikasi data, yaitu Random Forest, XGBoost (Extreme Gradient Boosting), SVM (Support Vector Machine), KNN (K-Nearest Neighbour), dan Logistic Regression. Pemilihan kelima algoritma ini didasarkan pada karakteristiknya yang beragam, mulai dari model berbasis ensemble, model linear, hingga model berbasis jarak, sehingga diharapkan mampu memberikan perbandingan performa yang komprehensif dalam memprediksi status banjir berdasarkan data ketinggian air.

Selain menggunakan algoritma *machine learning* konvensional, penelitian ini juga mengimplementasikan metode *deep learning* Long Short Term Memory (LSTM) untuk menangkap pola temporal pada data hidrologi secara berkala. LSTM merupakan pengembangan dari Recurrent Neural Network (RNN) yang dirancang untuk mempelajari dependensi jangka panjang pada data sekuensial melalui mekanisme *memory cell* dan *gate*. Penggunaan LSTM dalam penelitian ini didasarkan pada karakteristik data ketinggian air

yang bersifat *time series* dan memiliki ketergantungan temporal antar waktu pengamatan. Perubahan ketinggian air pada suatu waktu dipengaruhi oleh kondisi sebelumnya sehingga pendekatan *sequence learning* dinilai lebih sesuai dibandingkan model *machine learning* statis.

Setiap model dilatih menggunakan hasil yang telah melalui tahap EDA dan preprocessing. Proses pelatihan dilakukan pada data training, kemudian dievaluasi menggunakan data testing untuk mengukur kemampuan generalisasi model terhadap data baru. Parameter dasar digunakan sebagai baseline, dengan beberapa penyesuaian pada parameter tertentu untuk meningkatkan performa model. Hasil dari masing – masing algoritma kemudian dibandingkan menggunakan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score* untuk menentukan model terbaik dalam prediksi status banjir.

#### E. Hyperparameter Tuning

Hyperparameter tuning merupakan tahapan penting dalam proses pemodelan dalam meningkatkan performa model *machine learning*. Pada penelitian ini, teknik hyperparameter tuning dilakukan menggunakan model *grid search*. *Grid search* bekerja dengan cara menguji seluruh kombinasi parameter yang telah ditentukan sebelumnya secara sistematis untuk menemukan kombinasi terbaik yang menghasilkan performa model yang paling optimal [21]. Proses ini dilakukan menggunakan fungsi *GridSearchCV* dari library *scikit-learn*, yang juga mengintegrasikan teknik *cross-validation* untuk memastikan bahwa model tidak mengalami *overfitting* terhadap data latih.

*GridSearchCV* digunakan untuk menguji berbagai kombinasi parameter guna memperoleh konfigurasi model terbaik. Hyperparameter tuning dilakukan menggunakan 5-fold *cross-validation* karena penelitian ini berfokus pada klasifikasi status banjir saat pengamatan (*snapshot classification*), bukan peramalan deret waktu. Korelasi antar stasiun memungkinkan model memanfaatkan informasi spasial yang berkontribusi terhadap peningkatan akurasi prediksi. Tuning dilakukan pada parameter utama masing-masing model, seperti jumlah estimator dan kedalaman pohon pada Random Forest dan XGBoost, parameter C dan gamma pada SVM, jumlah tetangga pada KNN, serta parameter regularisasi pada Logistic Regression. Pada LSTM, tuning mencakup jumlah neuron, dropout rate, batch size, dan epoch, dengan *EarlyStopping* untuk mengurangi risiko *overfitting*. Konfigurasi terbaik kemudian dipilih berdasarkan nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

#### F. Evaluasi

Evaluasi model dilakukan pada data uji (*test*) yang tidak pernah digunakan selama pelatihan. Beberapa metrik evaluasi digunakan untuk menilai performa masing – masing model secara komprehensif, meliputi *accuracy*, *precision*, *recall*, *F1-score*, serta *confusion matrix*.

*Accuracy* mengukur proporsi prediksi yang benar secara keseluruhan. *Classification Report* menyajikan *precision*,

*recall*, *F1-score* untuk setiap kelasnya secara terperinci. *Confusion matrix* divisualisasikan menggunakan heatmap untuk memberikan gambaran langsung mengenai jumlah prediksi benar dan salah per kelas, sehingga memudahkan mengidentifikasi jenis kesalahan yang paling sering terjadi pada masing – masing model. Perbandingan ketiga model dilakukan berdasarkan nilai *accuracy* dan *F1-score* untuk menentukan algoritma yang paling sesuai dalam memprediksi status banjir berdasarkan data ketinggian air dari post pemantauan.

a. *Accuracy*

*Accuracy* merupakan proporsi prediksi benar secara keseluruhan yang disajikan dalam rumus (1)

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (1)$$

b. *Precision*

*Precision* merupakan ketepatan prediksi positif yang disajikan dalam rumus (2).

$$Precision = \frac{TP}{(TP+FP)} \quad (2)$$

c. *Recall*

*Recall* merupakan sensitivitas untuk mendeteksi kelas positif.

$$Recall = \frac{TP}{(TP+FN)} \quad (3)$$

d. *F1-score*

*F1-score* merupakan nilai yang diperoleh untuk menilai apakah nilai precision dan recall atau tidak dengan melakukan perbandingan di antara keduanya seperti yang terdapat pada persamaan[13].

$$F1 - score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

### III. HASIL DAN PEMBAHASAN

#### A. Dataset

Dataset yang digunakan dalam penelitian ini merupakan data historis ketinggian air dari tujuh pos pemantauan di sepanjang aliran Sungai Ciliwung, yaitu Katulampa, Pos Depok, Manggarai, Istiqlal, Jembatan Merah, Flusing Ancol, dan Marina Ancol. Data ini diperoleh dari platform kaggle dalam format CSV dan dimuat ke lingkungan komputasi menggunakan library pandas melalui Google Drive. Sungai Ciliwung dipilih sebagai objek penelitian karena merupakan sungai utama yang melintasi wilayah DKI Jakarta dan menjadi salah satu penyebab banjir.

Dataset yang digunakan terdiri dari 616 record data historis ketinggian muka air dari tujuh pos pemantauan Sungai Ciliwung. Jumlah dataset yang relatif terbatas disebabkan

oleh keterbatasan ketersediaan data publik banjir perkotaan di Indonesia yang telah terlabel secara lengkap. Sebagian besar data hidrologi di Indonesia masih bersifat institusional dan tidak tersedia secara terbuka untuk keperluan penelitian. Oleh karena itu, dataset publik yang tersedia melalui Kaggle dipilih karena telah menyediakan data multi-stasiun yang merepresentasikan kondisi hidrologi perkotaan Jakarta secara berkelanjutan. Meskipun jumlah data relatif kecil dibandingkan penelitian internasional, dataset ini tetap relevan karena berasal dari wilayah yang memiliki tingkat kerawanan banjir tinggi dan telah digunakan pada beberapa penelitian prediksi banjir sebelumnya.

TABEL 1  
INFORMASI UMUM DATASET

Atribut	Keterangan
Jumlah Fitur	10 Kolom ( 7 Fitur Ketinggian Air, Tanggal, Waktu, Status Banjir )
Fitur Input	Katulampa, Pos Depok, Manggarai, Istiqlal, Jembatan Merah, Flusing Ancol, Marina Ancol
Variabel Target	Status Banjir (0 = tidak banjir, 1 = banjir )
Fitur Temporal	Hour, Day, Month yang diekstraksi dari Tanggal dan Waktu
Tipe Data Fitur	Numerik (float64)

Gambaran umum struktur dataset setelah pemuatan awal disajikan dalam Tabel 1. Dataset terdiri dari 10 kolom yang mencakup kolom identitas waktu (Tanggal dan Waktu), tujuh kolom fitur ketinggian air dari masing – masing pos pemantauan, serta satu kolom target yaitu Status Banjir. Setiap banjir mempresentasikan satu rekaman pengukuran berkala yang mencerminkan kondisi hidrologi sungai pada waktu tertentu.

Variabel target Status Banjir merupakan label klasifikasi biner yang terdiri atas kelas 0 (tidak banjir) dan 1 (banjir). Klasifikasi tersebut mengacu pada indikator kejadian banjir berdasarkan tinggi muka air. Menurut KMS BMKG[22], suatu wilayah dapat dikategorikan mengalami banjir apabila ketinggian genangan air melebihi 40 cm, sedangkan genangan dengan ketinggian di bawah ambang tersebut tidak dikategorikan sebagai banjir. Oleh karena itu, label yang digunakan dalam dataset merepresentasikan kondisi hidrologi berdasarkan ambang batas ketinggian air yang berpotensi menimbulkan dampak banjir pada wilayah hilir.

#### B. Exploratory Data Analyst (EDA)

*Exploratory Data Analysis* dilakukan untuk memahami pola dan karakteristik data sebelum dilakukan pemodelan. Berikut hasil pengujian dalam tahap EDA dipaparkan pada subbab – subbabb berikut.

##### 1. Cek Informasi Umum Dataset

Pemeriksaan informasi umum dataset yang dilakukan menggunakan `df.info()` yang disajikan dalam Gambar 2 untuk mengetahui tipe data, jumlah kolom, dan jumlah baris pada

setiap fitur. Hasil pemeriksaan menunjukkan bahwa dataset terdiri dari 10 kolom Tanggal dan Waktu bertipe object.

```
<<class 'pandas.core.frame.DataFrame'>
RangeIndex: 616 entries, 0 to 615
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  ---          -
0   Tanggal         616 non-null   object
1   Waktu          616 non-null   object
2   Katulampa      616 non-null   int64
3   Pos Depok      616 non-null   int64
4   Manggarai      615 non-null   float64
5   Istiqlal       614 non-null   float64
6   Jembatan Merah 611 non-null   float64
7   Flusing Ancol  615 non-null   float64
8   Marina Ancol   615 non-null   float64
9   Status Banjir  616 non-null   int64
dtypes: float64(5), int64(3), object(2)
memory usage: 48.3+ KB
```

Gambar 2. Informasi Dataset

2. Cek Missing Value

Pemeriksaan informasi missing value menggunakan `df.isnull().sum()` untuk mengetahui nilai kosong pada dataset. Keberadaan missing value terdeteksi pada lima kolom ketinggian air yang tersaji dalam Gambar 3, yaitu Manggarai, Istiqlal, Jembatan Merah, Flusing Ancol dan Marina Ancol. Informasi ini menjadi acuan bagi penanganan missing value pada tahap preprocessing.

```
Ukuran dataset: (616, 10)
Missing values per kolom:
Tanggal          0
Waktu            0
Katulampa        0
Pos Depok        0
Manggarai        1
Istiqlal         2
Jembatan Merah   5
Flusing Ancol    1
Marina Ancol     1
Status Banjir    0
dtype: int64
```

Gambar 3. Cek Missing Value

3. Analisis Statistik Deskriptif

Analisis statistik deskriptif menggunakan `df.describe()` dilakukan untuk memperoleh gambaran menyeluruh tentang distribusi nilai pada masing – masing fitur numerik, mencakup nilai minimum, maksimum, rata – rata, standar deviasi, serta persentil ke-25, ke-50 (median), dan ke-75 yang tersaji dalam Gambar 4. Hasil analisis menunjukkan bahwa fitur – fitur ketinggian air memiliki rentang nilai yang cukup lebar dengan standar deviasi yang relatif besar, mengindikasikan adanya variasi kondisi ketinggian air yang signifikan sepanjang periode pencatatan. Hal ini konsisten dengan sifat data hidrologi yang sangat dipengaruhi oleh musim dan intensitas curah hujan. Kolom Katulampa secara khusus ditemukan memiliki nilai minimum negatif, yang secara fisik tidak mungkin terjadi dan perlu ditangani pada tahap preprocessing.

none	Katulampa	Pos Depok	Manggarai	Istiqlal	Jembatan Merah
count	616.000000	616.000000	615.000000	614.000000	611.000000
mean	41.170455	120.243506	608.827642	209.270358	174.101473
std	30.628461	43.269837	330.109905	53.347351	95.463595
min	-20.000000	65.000000	57.000000	85.000000	100.000000
25%	26.750000	96.000000	590.000000	170.000000	140.000000
50%	33.000000	111.000000	650.000000	211.000000	165.000000
75%	50.000000	140.000000	699.500000	240.000000	193.000000
max	240.000000	410.000000	6550.000000	336.000000	2290.000000

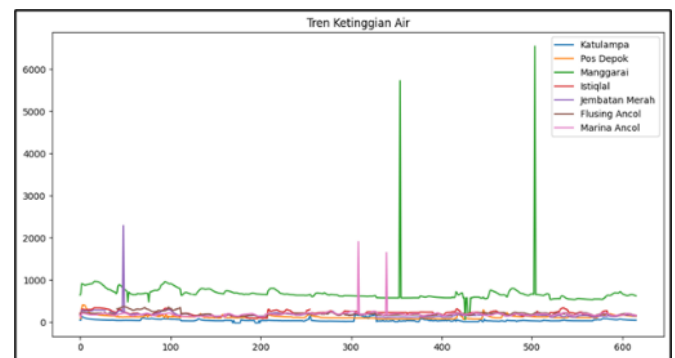
  

	Flusing Ancol	Marina Ancol	Status Banjir
count	615.000000	615.000000	616.000000
mean	197.284553	186.378862	0.527597
std	46.022251	94.915690	0.499644
min	115.000000	110.000000	0.000000
25%	170.000000	164.000000	0.000000
50%	189.000000	183.000000	1.000000
75%	210.000000	198.500000	1.000000
max	364.000000	1910.000000	1.000000

Gambar 4. Statistik Data

4. Analisis Tren Ketinggian Air

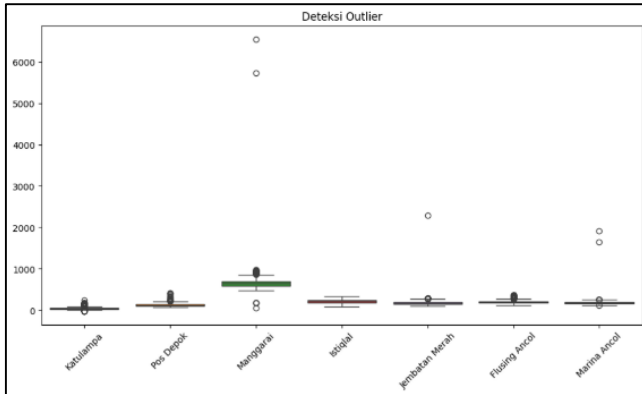
Analisis ketinggian air terhadap waktu dilakukan menggunakan line plot yang menampilkan perubahan ketinggian air dari seluruh tujuh pos pemantauan secara bersamaan dalam satu grafik pada Gambar 5. Visualisasi ini bertujuan untuk mengamati pola temporal dan dinamika hubungan antar post pemantauan. Hasil visualisasi menunjukkan adanya pola fluktuasi yang signifikan pada seluruh pos, dengan lonjakan periodik yang mencerminkan peristiwa curah hujan tinggi. Pos Katulampa menunjukkan lonjakan ketinggian yang secara temporal dari pada pos lainnya.



Gambar 5. Tren Ketinggian Air

5. Deteksi Outlier

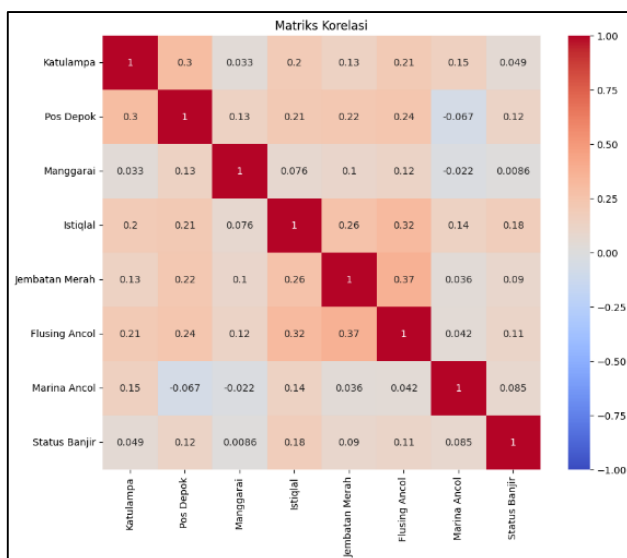
Deteksi outlier dilakukan menggunakan visualisasi boxplot untuk seluruh fitur ketinggian air. Boxplot merupakan alat diagnostik yang efektif mengidentifikasi nilai – nilai ekstrem yang berada di luar rentang interkuartil (IQR). Hasil visualisasi mengungkapkan adanya outlier ekstrem yang cukup banyak pada beberapa kolom fitur terutama Manggarai, Jembatan Merah, dan Marina Ancol. Outlier dengan nilai sangat tinggi pada kolom – kolom tersebut kemungkinan besar mempresentasikan kejadian banjir besar yang disebabkan ketinggian air yang jauh melampaui kondisi normal. Adanya outlier seperti yang tersaji dalam Gambar 6 menyebabkan model terlalu sensitif terhadap nilai – nilai ekstrem tersebut dan kurang mampu menggeneralisasi pada data baru. Oleh karena itu penanganan outlier diperlukan.



Gambar 6. Deteksi Outlier

6. Analisis Korelasi

Analisis korelasi antar fitur dilakukan menggunakan heatmap matriks korelasi untuk mengidentifikasi hubungan linear antara ketinggian air di setiap pos pemantauan satu sama lain dan korelasinya dengan variabel target yaitu Status Banjir. Nilai korelasi berkisar antar -1 (korelasi negatif yang sempurna) hingga +1 (korelasi positif sempurna), dengan nilai 0 menunjukkan tidak ada hubungan linear. Hasil heatmap menunjukkan bahwa secara umum terdapat korelasi positif yang kuat di antara seluruh fitur ketinggian air, yang mencerminkan bahwa ketinggian air di satu pos cenderung diikuti oleh peningkatan di pos lainnya akibat aliran hidrologi yang berkesinambungan. Korelasi antar pos yang berdekatan secara geografis seperti Manggarai – Istiqlal dan Istiqlal – Jembatan Merah tercatat paling tinggi. Tingginya korelasi antar stasiun menunjukkan bahwa kenaikan muka air pada wilayah hulu dapat menjadi indikator awal peningkatan muka air pada wilayah hilir. Informasi spasial ini memungkinkan model memanfaatkan keterkaitan antar lokasi sehingga meningkatkan akurasi prediksi status banjir.

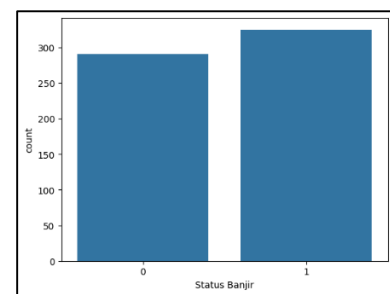


Gambar 7. Heatmap Korelasi

Penelitian ini menunjukkan bahwa pola kenaikan air pada stasiun lainnya secara berurutan. Oleh karena itu, penggunaan data multi stasiun memberikan informasi spasial yang lebih banyak dibandingkan salah satu titik pengamatan saja. Informasi spasial juga membantu model machine learning dan LSTM dalam mengenali pola distribusi banjir secara akurat.

7. Distribusi Kelas

Analisis distribusi kelas target menggunakan countplot menunjukkan adanya ketidakseimbangan kelas (class imbalance) yang cukup signifikan pada variabel Status Banjir. Kelas Banjir (1) mendominasi dataset dibandingkan kelas Tidak Banjir (0), yang merupakan kondisi yang umum ditemukan pada dataset pemantauan bencana alam karena secara alamiah kondisi normal jauh lebih sering terjadi dibandingkan kondisi bencana.



Gambar 8. Distribusi Target

Ketidakeimbangan yang ditunjukkan pada gambar 8 ini berpotensi menyebabkan model machine learning bersifat bias terhadap kelas mayoritas, menghasilkan prediksi yang cenderung mengklasifikasi semua sampel sebagai tidak banjir untuk memaksimalkan akurasi keseluruhan. Kondisi ini tentu sangat tidak diinginkan dalam konteks sistem peringatan dini banjir karena kegagalan mendeteksi banjir yang sesungguhnya terjadi (false negative) dapat berakibat fatal.

C. Hasil Preprocessing

Berdasarkan temuan pada fase EDA, proses preprocessing dilakukan secara bertahap dan berurutan untuk memastikan data yang digunakan dalam pemodelan bersih, konsisten, dan berada pada skala yang sesuai. Setiap langkah preprocessing dirancang untuk menangani permasalahan spesifik yang teridentifikasi selama EDA. Berikut adalah uraian hasil dari setiap tahapan preprocessing yang dilakukan.

1. Penanganan Missing Value

Missing Values ditemukan pada lima kolom, yaitu Manggarai, Istiqlal, Jembatan Merah, Flusing Ancol, dan Marina Ancol. Kolom Katulampa dan Pos Depok tidak memiliki missing values. Strategi imputasi yang dipilih adalah pengisian dengan nilai median (nilai tengah) pada masing – masing kolom dengan median\_val = df[col].median(). Pemilihan median sebagai nilai imputasi didasarkan pada pertimbangan bahwa median lebih robust terhadap pengaruh nilai ekstrem dibandingkan rata – rata (mean), sehingga distribusi asli data tidak akan bergeser

secara signifikan akibat proses imputasi. Hal ini sangat relevan mengingat data ketinggian air bersifat right-skewed dan memiliki outlier yang cukup banyak.

```

... MENANGANI MISSING VALUES
-----
- Manggarai: 1 missing values + diisi dengan median (650.000000)
- Istiqal: 2 missing values + diisi dengan median (212.000000)
- Jembatan Merah: 5 missing values + diisi dengan median (166.000000)
- Flusing Ancol: 1 missing values + diisi dengan median (190.000000)
- Marina Ancol: 1 missing values + diisi dengan median (183.000000)
Missing values selesai ditangani
    
```

Gambar 9. Penanganan Missing Value

Setelah proses imputasi selesai dilakukan, pengecekan ulang menggunakan `df.isnull().sum()` mengonfirmasi bahwa tidak ada missing value lagi pada seluruh kolom dataset, dengan seluruh nilai menunjukkan angka 0. Sebagai bukti bahwa missing value telah ditangani telah disajikan dalam Gambar 9.

### 2. Penanganan Nilai Negatif

Nilai negatif pada kolom Katulampa berhasil diidentifikasi dan diganti menjadi 0 menggunakan fungsi `clip(lower=0)`. Langkah ini penting karena ketinggian air secara fisik tidak dapat bernilai negatif, sehingga nilai negatif yang ada merupakan artefak pengukuran yang tidak mencerminkan kondisi nyata. Penggantian dengan nilai 0 dipilih karena mempresentasikan kondisi ketinggian air yang berada pada atau di bawah titik referensi pengukuran

```

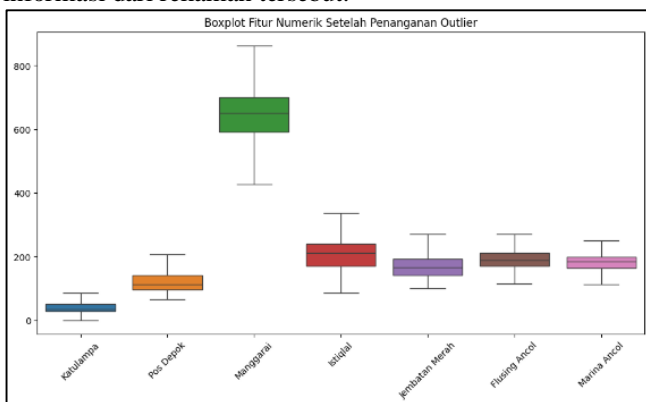
... MENANGANI NILAI NEGATIF
-----
- Katulampa: 13 nilai negatif + diganti dengan 0
Penanganan nilai negatif selesai
    
```

Gambar 10. Penanganan Nilai Negatif

### 3. Penanganan Nilai Outlier

Penanganan outlier dilakukan menggunakan metode IQR Capping dengan faktor 1,5. Metode ini diterapkan secara berurutan pada enam kolom, yaitu Manggarai, Jembatan Merah, Marina Ancol, Pos Depok, Flusing Ancol, dan Katulampa. Setiap kolom yang mengandung outlier di luar rentang

$[Q1 - 1,5 \times IQR, Q3 + 1,5 \times IQR]$  akan dipotong (*capped*) ke nilai batas yang sesuai, sehingga distribusi data menjadi lebih kompak tanpa sepenuhnya menghilangkan informasi dari rekaman tersebut.

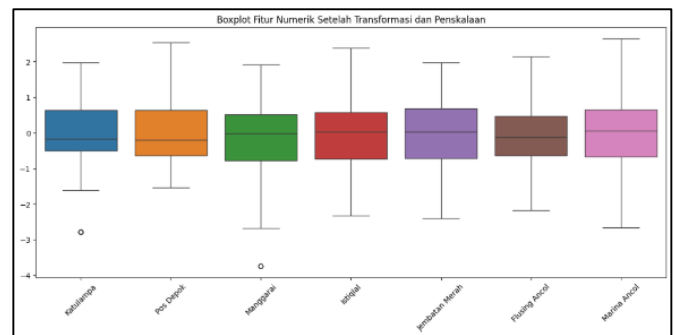


Gambar 11. Bloxpot Setelah Penanganan Outlier

Boxplot yang dibuat pada gambar 11 setelah proses *capping* menunjukkan distribusi yang jauh lebih rapi dibandingkan sebelumnya, dengan whisker yang lebih pendek dan jumlah titik di luar whisker yang lebih pendek dan jumlah whisker yang berkurang signifikan.

### 4. Transformasi Box-Cox dan Normalisasi

Transformasi Box-Cox diterapkan pada tiga kolom yang memiliki distribusi paling miring (*skewed*), yaitu Manggarai, Katulampa, dan Jembatan Merah. Transformasi ini bekerja dengan mencari parameter lambda ( $\lambda$ ) yang optimal secara otomatis untuk setiap kolom untuk menghasilkan distribusi yang paling mendekati distribusi normal. Nilai +1 ditambahkan pada setiap nilai sebelum transformasi diterapkan untuk memastikan tidak ada nilai nol atau negatif yang dapat menyebabkan indeterminate result pada fungsi logaritma. Histogram yang dibuat setelah transformasi memperlihatkan distribusi yang secara visual lebih simetris dan mendekati kurva normal dibandingkan sebelum transformasi diterapkan.



Gambar 12. Transformasi Box-Cox

Setelah transformasi, seluruh fitur numerik dinormalisasi menggunakan `StandardScaler` dari library `scikit-learn` yang ada pada gambar 12. Proses standarisasi mengubah setiap fitur sehingga memiliki rata-rata (*mean*) sebesar 0 dan standar deviasi sebesar 1. Normalisasi ini sangat penting terutama untuk model SVM yang sensitif terhadap perbedaan skala antar fitur. Tanpa normalisasi, fitur dengan rentang nilai yang lebih besar dapat secara tidak proporsional mendominasi proses pembelajaran model. Sebagai langkah dalam preprocessing, tiga fitur temporal baru (*Hour, Day, Month*) berhasil diekstraksi dari kolom *Tanggal dan Waktu* yang kemudian dihapus dari dataset.

Transformasi Box-Cox dipilih karena beberapa fitur memiliki distribusi right-skewed dengan nilai ekstrem yang cukup tinggi. Distribusi yang terlalu miring dapat menyebabkan model tertentu seperti SVM dan Logistic Regression bekerja kurang optimal. Setelah transformasi Box-Cox diterapkan, distribusi data menjadi lebih simetris sehingga meningkatkan stabilitas proses pembelajaran model dan membantu memperbaiki performa klasifikasi.

### 5. Adasyn Imbalance

Sebelum penerapan *adasyn*, dataset terlebih dulu dibagi menjadi data train sebesar 80% dan data test sebesar 20%

menggunakan fungsi `train_test_split` dengan `random_state=42` untuk menjamin reproduibilitas hasil.

TABEL 2.  
SUBNET TRAIN TEST

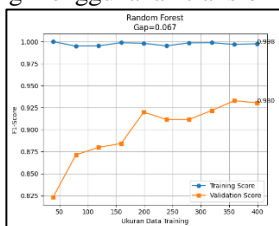
Subnet	Persentase (%)	Jumlah Data
Training	80	492
Testing	20	124

Pembagian ini menghasilkan dua subnet yang representatif pada Tabel 2. Data latih digunakan untuk melatih model, sedangkan data uji yang sama sekali tidak terlibat dalam proses pelatihan disimpan untuk keperluan evaluasi performa objektif dengan bukti pada Tabel 3.

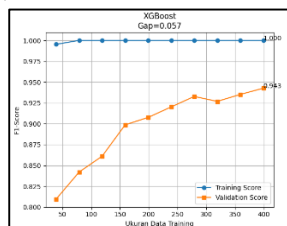
TABEL 3.  
PEMBAGIAN DATA

Subnet	Sebelum Adasyn	Sesudah Adasyn
Banjir	261	261
Tidak Banjir	231	261

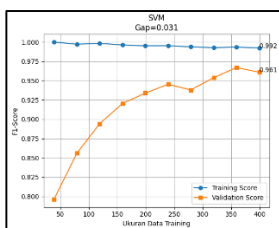
Adasyn kemudian diterapkan secara eksklusif hanya pada data latih (`X_train` dan `y_train`), bukan pada keseluruhan dataset. Parameter normalisasi hanya dipelajari dari data training menggunakan `fit()`, kemudian diterapkan pada data testing menggunakan `transform()`.



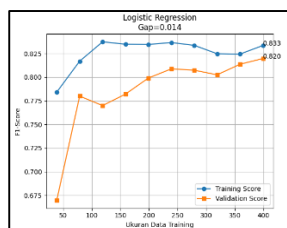
Gambar 13. Bebas Leakage Random Forest



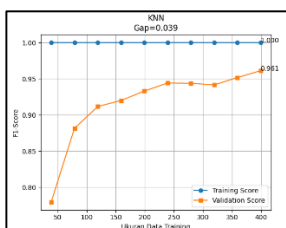
Gambar 14. Bebas Leakage XGBoost



Gambar 15. Bebas Leakage SVM



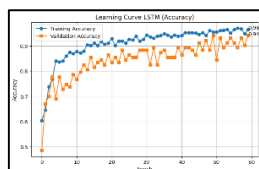
Gambar 16. Bebas Leakage Logistic Regression



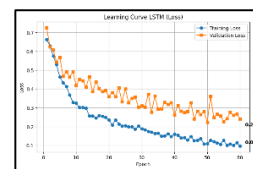
Gambar 17. Bebas Leakage KNN

Penerapan adasyn hanya pada data latih ini merupakan praktik yang benar dan sangat penting untuk mencegah data

leakage, yaitu situasi di mana informasi dari data uji secara tidak langsung ikut mempengaruhi proses pelatihan model, yang dapat menghasilkan estimasi performa yang terlalu optimistis dan tidak mencerminkan kemampuan generalisasi model pada data nyata dengan bukti pada gambar 13 hingga gambar 17. Hasil menunjukkan bahwa skor validasi meningkat secara bertahap seiring bertambahnya data training dan tidak langsung tinggi pada ukuran data kecil. Selain itu, gap antara skor training dan validasi masih berada dalam batas wajar sesuai karakteristik masing-masing model.



Gambar 18. Bukti Accuracy Bebas Leakage LSTM



Gambar 19. Bukti Bebas Leakage

Pada LSTM, Training Accuracy dan Validation Accuracy meningkat secara konsisten, sementara Training Loss dan Validation Loss menurun selama proses pelatihan. Pola tersebut menunjukkan bahwa model mempelajari pola data secara normal tanpa memanfaatkan informasi dari data testing. Dengan demikian, proses pembagian data, preprocessing, dan penerapan ADASYN telah dilakukan dengan benar sehingga hasil evaluasi yang diperoleh mencerminkan kemampuan generalisasi model yang sebenarnya.

#### D. Hyperparameter Tuning

Hyperparameter tuning dilakukan untuk memperoleh konfigurasi parameter terbaik pada masing-masing model sehingga performa klasifikasi dapat meningkat secara optimal. Proses tuning dilakukan menggunakan GridSearchCV pada model machine learning dan tuning manual pada model deep learning LSTM. Parameter yang dioptimalkan meliputi jumlah estimator, kedalaman pohon, parameter kernel, jumlah tetangga, parameter regularisasi, serta jumlah neuron dan dropout pada model LSTM.

Hasil tuning menunjukkan bahwa setiap model memiliki konfigurasi parameter optimal yang berbeda sesuai karakteristik algoritmanya. Pemilihan parameter yang tepat membantu model dalam meningkatkan kemampuan generalisasi serta mengurangi risiko overfitting pada data pengujian. Parameter optimal yang diperoleh pada masing-masing model disajikan pada Tabel 4.

Hyperparameter tuning dilakukan untuk memperoleh konfigurasi parameter terbaik pada masing-masing model sehingga performa klasifikasi dapat meningkat secara optimal. Proses tuning dilakukan menggunakan GridSearchCV pada model machine learning dan tuning manual pada model deep learning LSTM. Parameter yang dioptimalkan meliputi jumlah estimator, kedalaman pohon, parameter kernel, jumlah tetangga, parameter regularisasi, serta jumlah neuron dan dropout pada model LSTM.

Pada penelitian ini, proses tuning dilakukan setelah tahap preprocessing dan pembagian data training serta testing untuk menghindari terjadinya data leakage. Setiap kombinasi parameter diuji untuk mengetahui konfigurasi yang menghasilkan performa terbaik berdasarkan nilai accuracy, precision, recall, dan F1-score. Selain meningkatkan akurasi model, hyperparameter tuning juga bertujuan untuk meningkatkan kemampuan generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya.

Hasil tuning menunjukkan bahwa setiap model memiliki konfigurasi parameter optimal yang berbeda sesuai karakteristik algoritmanya. Pemilihan parameter yang tepat membantu model dalam menangkap pola data secara lebih efektif serta mengurangi risiko overfitting pada data pelatihan. Selain itu, tuning juga memberikan peningkatan performa yang cukup signifikan pada beberapa model, khususnya pada model SVM dan XGBoost.

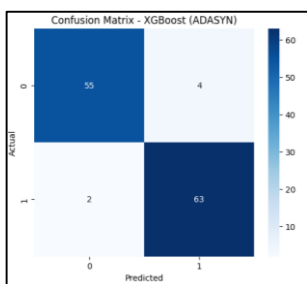
Parameter optimal yang diperoleh pada masing-masing model disajikan pada Tabel 4.

TABEL 4.  
PARAMETER OPTIMAL HYPERPARAMETER TUNING

Model	Parameter Optimal
Random Forest	n_estimators=200, max_depth=20, min_samples_split=2
XGBoost	learning_rate=0.1, max_depth=6, n_estimators=200
SVM	C=10, gamma=0.1, kernel='rbf'
KNN	n_neighbors=5, metric='euclidean'
Logistic Regression	C=1, solver='liblinear'
LSTM	units=128, dropout=0.3, batch_size=32, epochs=100

E. Evaluasi

Pada evaluasi disajikan hasil evaluasi lima algoritma machine learning, yaitu Random Forest, XGBoost, SVM, KNN, dan Logistic Regression, serta metode deep learning Long Short-Term Memory (LSTM) dalam memprediksi status banjir berdasarkan data ketinggian air. Evaluasi dilakukan pada kondisi sebelum dan sesudah hyperparameter tuning menggunakan GridSearchCV untuk model machine learning dan tuning manual pada model LSTM guna



Gambar 22. Confusion Matrix XGBoost Sebelum Hyperparameter Tuning



Gambar 23. Confusion Matrix XGBoost Sesudah Hyperparameter Tuning

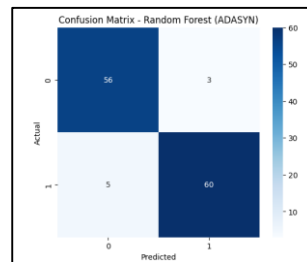
mengetahui pengaruh optimasi parameter terhadap performa model. Setiap model dievaluasi menggunakan accuracy,

precision, recall, F1-score, dan confusion matrix untuk menganalisis pola kesalahan prediksi.

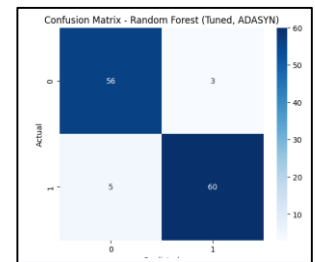
Sebelum proses pemodelan, dataset telah melalui tahapan preprocessing meliputi penanganan missing value, outlier, transformasi Box-Cox, normalisasi, ekstraksi fitur temporal, dan penanganan imbalance data menggunakan adasyn. Hasil evaluasi menunjukkan adanya perbedaan performa antar model, di mana XGBoost dan LSTM menunjukkan performa yang kompetitif. Selain itu, beberapa model mengalami peningkatan performa yang signifikan setelah dilakukan hyperparameter tuning.

1. Random Forest

Model Random Forest dilatih dengan parameter n\_estimators=100, max\_depth=None, dan random\_state=42. Confusion matrix pada Gambar 20 dan Gambar 21 menunjukkan bahwa model ini menghasilkan nilai True Positive (TP) dan True Negative (TN) yang dominan, mengindikasikan kemampuan model dalam mengklasifikasikan kedua kelas secara konsisten. Nilai False Positive (FP) yang rendah menunjukkan bahwa model jarang salah memprediksi kondisi tidak banjir sebagai banjir, sementara False Negative (FN) yang masih muncul dalam jumlah kecil mencerminkan adanya sebagian kasus banjir yang tidak berhasil terdeteksi. Pola confusion matrix sebelum dan sesudah hyperparameter tuning tidak menunjukkan perubahan, mengindikasikan bahwa distribusi kesalahan prediksi model sudah stabil sejak konfigurasi awal.



Gambar 20. Confusion Matrix RF Sebelum Hyperparameter Tuning



Gambar 21. Confusion Matrix RF Sesudah Hyperparameter Tuning

Nilai FN yang rendah sangat krusial dalam konteks sistem peringatan dini banjir, karena kegagalan mendeteksi kejadian banjir yang sesungguhnya memiliki dampak yang jauh lebih berbahaya dibandingkan alarm palsu. Selain itu, jumlah false positive yang tidak terlalu tinggi menunjukkan bahwa model cukup stabil dalam membedakan kondisi banjir dan tidak banjir.

2. XGBoost

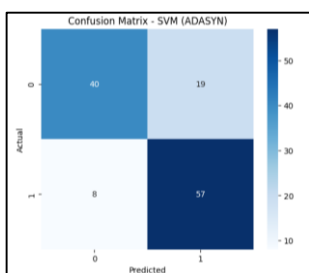
Model XGBoost dilatih menggunakan XGBClassifier dengan parameter random\_state=42 dan eval\_metric='logloss'. XGBoost memanfaatkan pendekatan gradient boosting yang membangun pohon secara sekuensial, di mana setiap iterasi berfokus pada koreksi kesalahan dari iterasi sebelumnya.

Confusion matrix pada Gambar 22 dan Gambar 23 memperlihatkan bahwa XGBoost menghasilkan nilai TP yang

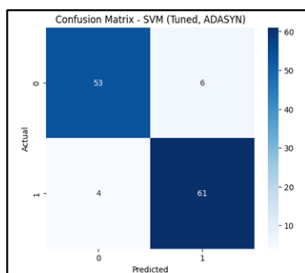
sangat tinggi dengan FN yang sangat rendah, baik sebelum maupun sesudah hyperparameter tuning. Hal ini menunjukkan bahwa hampir seluruh kejadian banjir berhasil terdeteksi dengan benar oleh model. Nilai FP yang sedikit lebih besar dibanding FN mengindikasikan model sedikit lebih cenderung memprediksi banjir daripada melewatkannya, yang merupakan perilaku yang lebih aman dalam konteks sistem peringatan dini. Setelah tuning, jumlah FP terlihat berkurang sehingga distribusi kesalahan prediksi menjadi lebih seimbang

### 3. SVM

Model SVM (Support Vector Machine) dilatih menggunakan SVC dengan kernel RBF (Radial Basis Function, default), probability=True, dan random\_state=42. Kernel RBF dipilih karena mampu memetakan data ke ruang fitur berdimensi lebih tinggi secara implisit melalui fungsi kernel, sehingga dapat memisahkan kelas yang tidak dapat dipisahkan secara linear di ruang fitur aslinya. Parameter probability=True diaktifkan agar model dapat menghasilkan estimasi probabilitas selain prediksi kelas biner.



Gambar 24. Confusion Matrix SVM Sebelum Hyperparameter Tuning

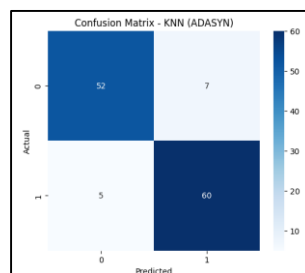


Gambar 25. Confusion Matrix SVM Sesudah Hyperparameter Tuning

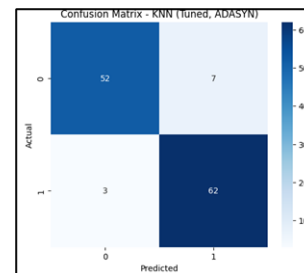
Confusion matrix pada Gambar 24 menunjukkan bahwa sebelum tuning, model SVM memiliki jumlah FP yang cukup besar, artinya banyak kondisi tidak banjir yang salah diklasifikasikan sebagai banjir. Setelah hyperparameter tuning seperti yang ditunjukkan Gambar 25, distribusi confusion matrix membaik secara signifikan dengan penurunan FP yang nyata, sementara FN juga turut berkurang, sehingga TP dan TN menjadi lebih dominan dan model menunjukkan kemampuan klasifikasi yang jauh lebih seimbang antara kedua kelas.

### 4. KNN

Model K-Nearest Neighbour (KNN) digunakan sebagai metode klasifikasi berbasis kedekatan jarak antar data. KNN bekerja dengan menentukan kelas suatu data berdasarkan mayoritas kelas dari k tetangga terdekatnya dalam ruang fitur. Pada penelitian ini, model KNN diimplementasikan menggunakan parameter jumlah tetangga (k) yang ditentukan melalui eksperimen, serta menggunakan metrik jarak Euclidean. Data yang telah dinormalisasi sebelumnya sangat penting dalam penggunaan KNN agar perhitungan jarak antar fitur tidak bias terhadap skala tertentu.



Gambar 26. Confusion Matrix KNN Sebelum Hyperparameter Tuning

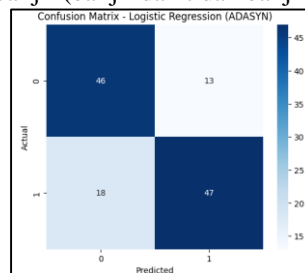


Gambar 27. Confusion Matrix KNN Sesudah Hyperparameter Tuning

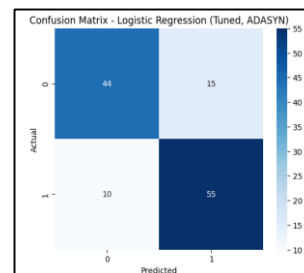
Confusion matrix pada Gambar 26 menunjukkan bahwa sebelum tuning, model KNN memiliki jumlah FP yang sedikit lebih besar dibandingkan FN, artinya model lebih cenderung salah mengklasifikasikan kondisi tidak banjir sebagai banjir daripada melewatkan kejadian banjir yang nyata. Setelah tuning seperti terlihat pada Gambar 27, jumlah FN berhasil ditekan lebih lanjut dengan konsekuensi FP yang sedikit meningkat, sehingga model menjadi lebih sensitif dalam mendeteksi kelas banjir sebuah trade-off yang dapat diterima mengingat prioritas sistem peringatan dini adalah meminimalkan kejadian banjir yang tidak terdeteksi.

### 5. Logistic Regression

Model Logistic Regression digunakan sebagai baseline model dalam penelitian ini karena sifatnya yang sederhana dan mudah diinterpretasikan. Logistic Regression bekerja dengan memodelkan probabilitas suatu data termasuk ke dalam kelas tertentu menggunakan fungsi sigmoid[23]. Pada penelitian ini, Logistic Regression diimplementasikan menggunakan parameter default dengan optimasi solver yang sesuai untuk dataset berukuran sedang. Model ini sangat efektif untuk kasus klasifikasi biner seperti prediksi status banjir (banjir dan tidak banjir).



Gambar 28. Confusion Matrix Logistic Regression Sebelum Hyperparameter Tuning



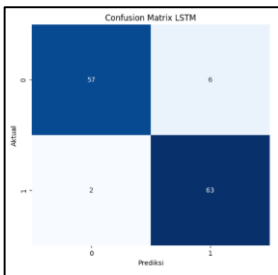
Gambar 29. Confusion Matrix Logistic Regression Sesudah Hyperparameter Tuning

Confusion matrix pada Gambar 28 menunjukkan bahwa sebelum tuning, model Logistic Regression memiliki jumlah FN yang relatif besar dibandingkan model lain, mengindikasikan cukup banyak kejadian banjir yang gagal terdeteksi. Setelah tuning seperti terlihat pada Gambar 29, jumlah FN berkurang meski FP sedikit meningkat. Secara keseluruhan, distribusi TP dan TN pada model ini masih lebih rendah dibandingkan model-model lainnya, mencerminkan

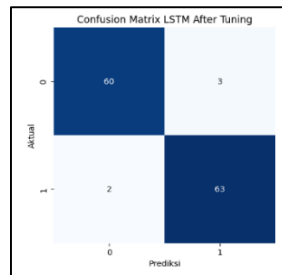
keterbatasan model linear dalam memisahkan pola data ketinggian air yang bersifat non-linear.

## 6. LSTM

Model LSTM digunakan untuk menangkap pola temporal pada data ketinggian air yang bersifat time-series. Berbeda dengan model machine learning konvensional, LSTM mampu menyimpan informasi jangka panjang melalui mekanisme memory cell sehingga lebih sesuai untuk data hidrologi yang memiliki ketergantungan waktu.



Gambar 30. Confusion Matrix LSTM Sebelum Hyperparameter Tuning



Gambar 31. Confusion Matrix LSTM Setelah Hyperparameter Tuning

Confusion matrix pada Gambar 30 menunjukkan bahwa sebelum tuning, LSTM sudah menghasilkan distribusi TP dan TN yang tinggi dengan FP dan FN yang relatif kecil dan seimbang. Setelah tuning seperti terlihat pada Gambar 31, distribusi confusion matrix semakin baik dengan FP dan FN yang sama-sama berkurang secara merata. Rendahnya FN pada LSTM menjadikan model ini sangat andal untuk sistem peringatan dini, karena hampir seluruh kejadian banjir berhasil teridentifikasi dengan benar tanpa lonjakan FP yang signifikan.

## F. Perbandingan Evaluasi Model

TABEL 5.

PERBANDINGAN MODEL

RF	Accuracy	Precision	Recall	F1-score
Sebelum	0.937	0.940	0.935	0.935
Sesudah	0.945	0.945	0.945	0.945
<b>XGB</b>				
Sebelum	0.952	0.940	0.969	0.955
Sesudah	0.960	0.955	0.969	0.962
<b>SVM</b>				
Sebelum	0.782	0.750	0.877	0.809
Sesudah	0.919	0.910	0.938	0.924
<b>LR</b>				
Sebelum	0.750	0.783	0.723	0.752
Sesudah	0.798	0.786	0.84	0.815
<b>KNN</b>				
Sebelum	0.903	0.895	0.923	0.909
Sesudah	0.919	0.899	0.954	0.925
<b>LSTM</b>				
Sebelum	0.937	0.940	0.935	0.935
Sesudah	0.945	0.945	0.945	0.945

Berdasarkan Tabel 5, perbandingan accuracy seluruh model sebelum dan sesudah hyperparameter tuning menunjukkan perbedaan performa yang cukup signifikan. XGBoost mencatat accuracy tertinggi yaitu 96,0% setelah tuning, menjadikannya model terbaik dalam penelitian ini. LSTM menghasilkan accuracy 94,5% setelah tuning dan bersaing ketat dengan Random Forest yang stabil di angka 93,5% tanpa perubahan sebelum maupun sesudah tuning, mengindikasikan bahwa konfigurasi default Random Forest sudah mendekati optimal pada dataset ini. KNN dan SVM sama-sama mencapai accuracy 91,9% setelah tuning, di mana SVM menunjukkan peningkatan paling dramatis yakni dari 78,2% menjadi 91,9% (selisih 13,7 poin persentase), membuktikan bahwa pemilihan hyperparameter yang tepat sangat krusial bagi model ini. Logistic Regression berada di posisi terendah dengan accuracy 79,8% setelah tuning, mencerminkan keterbatasan model linear dalam menangkap hubungan non-linear antara data ketinggian air dan status banjir. Secara keseluruhan, model berbasis ensemble (XGBoost dan Random Forest) serta deep learning (LSTM) terbukti lebih unggul dan stabil dibandingkan model lainnya, sementara hyperparameter tuning memberikan manfaat paling besar pada model yang sensitif terhadap konfigurasi parameter seperti SVM dan Logistic Regression.

Hasil penelitian ini juga sejalan dengan berbagai penelitian terdahulu yang menerapkan machine learning untuk prediksi dan klasifikasi banjir pada berbagai wilayah. Rahmadani et al. di Palembang menunjukkan bahwa Random Forest mampu memberikan performa yang baik dalam prediksi curah hujan dengan nilai koefisien determinasi ( $R^2$ ) sebesar 74,84% [6]. Penelitian Oemarki et al. di DKI Jakarta menunjukkan bahwa algoritma Support Vector Machine (SVM) mampu menghasilkan akurasi rata-rata sebesar 90,19% dan lebih unggul dibandingkan K-Nearest Neighbour (KNN) [9]. Penelitian Bagaskara dan Hartomo di Kota Semarang memperoleh akurasi sebesar 85% menggunakan algoritma KNN [24]. Semen Penelitian serupa juga dilaporkan oleh Suaif dan Rahayu di Bandar Lampung [16], di mana XGBoost menghasilkan performa terbaik dibandingkan ARIMA dan Random Forest. Penelitian Evitasari et al. di Samarinda juga membuktikan bahwa optimasi Genetic Algorithm pada SVM dapat meningkatkan akurasi klasifikasi banjir dari 52,71% menjadi 66,16% [25]. Pada tingkat internasional, penelitian Zuhairi et al. [26] di Malaysia menunjukkan bahwa model berbasis ensemble seperti Random Forest dan XGBoost memiliki performa yang lebih unggul dibandingkan beberapa algoritma konvensional dalam tugas flood forecasting. Dibandingkan dengan penelitian-penelitian tersebut, XGBoost pada penelitian ini berhasil mencapai accuracy sebesar 96,0%, yang menunjukkan bahwa model tersebut mampu memberikan performa yang sangat kompetitif dalam klasifikasi status banjir berdasarkan data ketinggian muka air.

#### IV. KESIMPULAN

Penelitian ini membandingkan lima model machine learning, yaitu Random Forest, XGBoost, SVM, KNN, dan Logistic Regression, untuk memprediksi keadaan banjir berdasarkan data tinggi air dari tujuh pos pengamatan di DKI Jakarta. Berdasarkan hasil penelitian dan penilaian yang telah dilaksanakan, dapat diambil beberapa kesimpulan sebagai berikut.

Berdasarkan hasil penelitian yang telah dilakukan, penerapan algoritma machine learning dan deep learning untuk klasifikasi status banjir berdasarkan data ketinggian muka air mampu menghasilkan performa yang baik. Dari enam model yang diuji, yaitu Random Forest, XGBoost, K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Logistic Regression, dan Long Short-Term Memory (LSTM), model XGBoost menunjukkan performa terbaik setelah proses hyperparameter tuning dengan accuracy sebesar 96,0%, precision sebesar 96,1%, recall sebesar 96,0%, dan F1-score sebesar 96,2%. Hasil tersebut menunjukkan bahwa XGBoost memiliki kemampuan yang sangat baik dalam mengidentifikasi pola hubungan antara data ketinggian muka air dan status banjir dibandingkan model lainnya

Proses hyperparameter tuning terbukti mampu meningkatkan performa pada sebagian besar model, terutama SVM yang mengalami peningkatan accuracy dari 78,2% menjadi 91,9%. Sementara itu, Random Forest menunjukkan performa yang relatif stabil sebelum dan sesudah tuning, yang mengindikasikan bahwa konfigurasi awal model tersebut telah mendekati kondisi optimal untuk dataset yang digunakan. Selain itu, model LSTM berhasil mencapai accuracy sebesar 94,5%, menunjukkan bahwa pendekatan deep learning juga mampu memberikan hasil yang kompetitif dalam klasifikasi status banjir.

Validasi menggunakan learning curve menunjukkan bahwa seluruh model memiliki kemampuan generalisasi yang baik dan tidak ditemukan indikasi data leakage yang signifikan. Pada model machine learning, selisih antara skor training dan validation relatif kecil, sedangkan pada model LSTM kurva training dan validation menunjukkan pola yang stabil tanpa peningkatan validation loss yang signifikan pada akhir proses pelatihan. Hal ini menunjukkan bahwa proses pembagian data, normalisasi, oversampling menggunakan ADASYN, serta hyperparameter tuning telah dilakukan dengan benar sehingga performa model yang diperoleh dapat merepresentasikan kemampuan model terhadap data yang belum pernah dilihat sebelumnya.

Hasil penelitian ini juga sejalan dengan berbagai penelitian terdahulu yang menerapkan machine learning untuk prediksi dan klasifikasi banjir di berbagai wilayah. Temuan tersebut menunjukkan bahwa pendekatan berbasis machine learning, khususnya algoritma ensemble seperti XGBoost dan Random Forest, memiliki potensi untuk diterapkan dalam sistem peringatan dini banjir guna mendukung pengambilan keputusan yang lebih cepat dan akurat.

Namun demikian, penelitian ini masih memiliki beberapa keterbatasan. Evaluasi model dalam penelitian ini masih berfokus pada metrik performa klasifikasi seperti accuracy, precision, recall, dan F1-score, sehingga belum mencakup pengujian yang lebih luas terhadap ketahanan model dalam berbagai kondisi operasional. Selain itu, model dibangun dan diuji menggunakan data dari wilayah penelitian yang sama sehingga kemampuan generalisasi model terhadap wilayah lain dengan karakteristik hidrologi, topografi, pola curah hujan, serta sistem drainase yang berbeda belum dapat dipastikan secara langsung. Penelitian ini juga belum mengevaluasi performa model pada kondisi ekstrem atau data yang berada di luar distribusi data pelatihan (*out-of-distribution events*), seperti kejadian banjir yang memiliki pola berbeda dari data historis yang digunakan. Oleh karena itu, penelitian selanjutnya disarankan untuk melakukan validasi menggunakan dataset dari wilayah yang berbeda, menguji model pada kondisi ekstrem, serta mengevaluasi performa model pada data yang lebih beragam agar diperoleh model yang lebih robust dan memiliki kemampuan generalisasi yang lebih baik.

Secara keseluruhan, penelitian ini menunjukkan bahwa algoritma XGBoost merupakan model yang lebih efektif untuk klasifikasi status banjir berdasarkan data ketinggian muka air pada dataset yang digunakan, serta memiliki potensi untuk mendukung pengembangan sistem peringatan dini banjir yang lebih cepat, akurat, dan adaptif dalam upaya mitigasi risiko bencana banjir.

#### DAFTAR PUSTAKA

- [1] M. A. Hasanah, S. Soim, and A. S. Handayani, "Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir," vol. 5, no. 2, 2021.
- [2] J. Akbar, M. Ali, and S. Yudono, "Water Level Classification for Detect Flood Disaster Status using KNN and SVM," vol. 13, pp. 298–302, 2024.
- [3] M. E. El-mahdy, F. Ali, F. Ibraheem, A. Fakhry, and A. El-tantawi, "Flood classification and prediction in South Sudan using artificial intelligence models under a changing climate," *Alexandria Eng. J.*, vol. 97, no. March, pp. 127–141, 2024, doi: 10.1016/j.aej.2024.03.082.
- [4] M. D. Wilson, E. M. Lane, and J. Brasington, "Estimating uncertainty in flood model outputs using machine learning informed by Monte Carlo analysis," *J. Hydrol.*, vol. 662, no. PC, p. 133928, 2025, doi: 10.1016/j.jhydrol.2025.133928.
- [5] Y. Tang, Y. Sun, Z. Han, S. Soomro, and Q. Wu, "Journal of Hydrology: Regional Studies flood forecasting based on machine learning pattern recognition and dynamic migration of parameters," *J. Hydrol. Reg. Stud.*, vol. 47, p. 101406, 2023, doi: 10.1016/j.ejrh.2023.101406.
- [6] N. Rahmadani, A. S. Handayani, and I. Hadi, "Penerapan Algoritma Random Forest untuk Memprediksi Curah Hujan pada Masa Mendatang di Daerah Berpotensi Banjir," vol. 6, no. 2, pp. 1222–1230, 2024, doi: 10.47065/bits.v6i2.5593.
- [7] V. Frendyatha, M. Akrom, and G. Alfa, "Investigasi Efisiensi Penghambatan Korosi Senyawa Quinoxaline Berbasis Machine Learning A Study on the Corrosion Inhibition Efficiency of Quinoxaline Compounds Utilizing Machine Learning," vol. 21, no. 2, pp. 65–69, 2024.
- [8] S. Samantaray, A. Sahoo, and A. Agnihotri, "MethodsX Prediction of Flood Discharge Using Hybrid PSO-SVM Algorithm in Barak

- River Basin,” *MethodsX*, vol. 10, no. February, p. 102060, 2023, doi: 10.1016/j.mex.2023.102060.
- [9] M. F. Oemarki *et al.*, “Perbandingan Akurasi Metode Support Vector Machine Dan K-Nearest Neighbour Dalam Prediksi Curah Hujan,” no. April, pp. 160–167, 2024.
- [10] C. Mondal and J. Uddin, “Heliyon Classification of short-term flood events using stochastic variable selection and Gaussian Naïve Bayes classifier : A case study of Sirajganj district , Bangladesh,” vol. 11, no. October 2024, 2025.
- [11] S. Cumel, David Zamri, Rahmaddeni, “Perbandingan Metode Data Mining untuk Prediksi Banjir dengan Algoritma Naïve Bayes dan KNN,” *SENTIMAS Semin. Nas. Penelit. dan ...*, pp. 40–48, 2022, [Online]. Available: <https://journal.irpi.or.id/index.php/sentimas/article/download/353/132>
- [12] S. M. Natzir, “Perbandingan Kinerja Model Pembelajaran Mesin dalam Prediksi Banjir menggunakan KNN , Naive Bayes , dan Random Forest,” vol. 14, no. c, pp. 59–64, 2023.
- [13] M. Bagas, A. Darmawan, F. Dewanta, and S. Astuti, “Analisis Perbandingan Algoritma Decision Tree , Random Forest , dan Naïve Bayes untuk Prediksi Banjir di Desa Dayeuhkolot Comparative Analysis of Decision Tree , Random Forest , and Naïve Bayes Algorithm for Flood Prediction at Dayeuhkolot Village,” vol. 9, no. 1, pp. 52–61.
- [14] D. R. Forest, W. H. Sasoko, E. W. Pujiharto, R. Haris, and A. Y. Kania, “Prediksi Banjir Di Dki Jakarta Dengan Menggunakan Algoritma K-Means,” vol. 05, no. 01, pp. 43–49, 2024.
- [15] B. Iklim and D. I. Indonesia, “Analisa perbandingan algoritma random forest dan naïve bayes untuk klasifikasi curah hujan berdasarkan iklim di indonesia,” vol. 9, no. 1, pp. 158–167, 2024.
- [16] D. I. Bandar, L. Menggunakan, and R. Forest, “Analisis faktor dan pola kejadian banjir di bandar lampung menggunakan arima, random forest, dan xgboost,” vol. 3, no. 2.
- [17] R. Handayan and W. Prarikeslan, “Analisis Multikriteria Dan Regresi Logistik Terhadap Kerawanan Banjir Di Kecamatan Koto Xi Tarusan, Kabupaten Pesisir Selatan,” vol. 10, no. September, pp. 240–262, 2025.
- [18] A. Wijayanto, A. Sugiharto, R. Santoso, U. Diponegoro, and P. Korespondensi, “Identifikasi Dini Curah Hujan Berpotensi Banjir Menggunakan Algoritma Long Short-Term Memory ( Lstm ) Dan Isolation Forest Early Identification Of Rainfall With Flood Potential Using Long Short-Term Memory ( Lstm ) And Isolation Forest Algorithms Case Stu,” vol. 11, no. 3, 2024, doi: 10.25126/jtiik.938718.
- [19] H. Kardhana, P. Dwi, R. Deno, F. Immaddudin, and W. Rohmat, “From unreliable observations to reliable forecasts : Enhancing Jakarta flood prediction using HEC-HMS-assisted LSTM modeling,” *Environ. Challenges*, vol. 23, no. December 2025, p. 101464, 2026, doi: 10.1016/j.envc.2026.101464.
- [20] T. Hermawan and E. Zuliarso, “Perbandingan Metode Recurrent Neural Network ( RNN ) dan Long Short-Term Memory ( LSTM ) untuk Prediksi Curah Hujan,” vol. 7, no. 2, pp. 1450–1463, 2025, doi: 10.47065/bits.v7i2.8099.
- [21] N. M. Lefi, M. Rahardi, F. I. Komputer, and U. A. Yogyakarta, “Hyperparameter Optimization and Feature Selection Analysis on the XGBoost Model for Hepatitis C Infection Prediction,” vol. 9, no. 6, pp. 3338–3345, 2025.
- [22] Nurhidayat, “Banjir, Rob, Genangan ( Penyebab, Dampak, dan Upaya Penanggulangan ),” kmsmbkg. [Online]. Available: <https://kms.bmkg.go.id/2025/11/banjir-rob-dan-genangan-penyebab-dampak-dan-upaya-penanggulangan/>
- [23] S. Jurnal, “(SINTEK) Analisis Sentimen Bencana Banjir Sumatera Menggunakan Tf-Idf Dan Logistic Regression,” vol. VI, no. 1.
- [24] F. Prone, A. Using, F. Cross, and N. Neighbors, “Klasifikasi Daerah Rawan Banjir menggunakan 10 - Fold Cross Validation dan K - Nearest Neighbors,” vol. 13, pp. 315–323, 2024.
- [25] Y. D. Evitasari, “Evaluasi Support Vector Machine Dengan Optimasi Metode Genetic Algorithm Pada Klasifikasi Banjir Kota Samarinda Evaluation Support Vector Machine With Optimization Genetic Algorithm Method On Flood Classification In Samarinda,” vol. 6, no. 1, pp. 49–53, 2023.
- [26] A. H. Zuhairi, F. Yakub, S. Member, and M. A. S. Omar, “Performance Analysis of Tree-Based Ensemble Machine Learning Model for Flood Forecasting in Tropical Regions,” *IEEE Access*, vol. 13, no. September, pp. 200840–200860, 2025, doi: 10.1109/ACCESS.2025.3632033.