# Penyusunan Model Untuk Predkisi Bencana Banjir Menggunakan Machine Learning

Rudi Hermawan[1], Asep Maulana[2], Arief Zulianto[3]

*Program Studi Informatika, Fakultas Teknik, Universitas Langlangbuana[1,2,3]*

[1]rudihermawan567@gmail.com

[2]siapok@yahoo.com

[3]madzul@unla.ac.id

*Abstrak*— ***Kecamatan Sukaresik, Tasikmalaya merupakan salah satu wilayah yang terdapat di hulu sungai Citanduy memiliki permasalahan banjir yang rutin terjadi. Banjir di wilayah ini terjadi pada saat musim penghujan dan rata – rata terjadi 3 kali dalam satu tahun, hal ini diakibatkan oleh luapan Sungai Citanduy. Untuk menciptakan sebuah sistem deteksi dini yang dapat digunakan untuk meprediksi terjadinya banjir, diperlukan beberapa tahapan analisis dan penelitian dari aspek atau variabel terjadinya banjir. Salah satu tahapan analisis yang diperlukan adalah proses pembelajaran pada data terdahulu variabel terjadinya banjir. Machine learning adalah salah satu cara yang dapat digunakan untuk mempelajari dan menguji data terdahulu dengan tujuan memperoleh sistem prediksi terbaik. Penelitian ini melakukan studi dan analisis untuk menerapkan teori machine learning dalam prediksi bencana banjir dengan data real yang didapatkan dari Balai Besar Wilayah Sungai Citanduy. Hasil dari penelitian ini didapatkan Algoritma terbaik yaitu Algoritma Decision Tree dengan akurasi hasil prediksi sebesar 98,29% dengan parameter, max_depth= 8, min_samples_split= 2, min_samples_leaf=2.***

***Kata kunci***— ***Artificial Neural Network, Decision Tree, K Nears Neighbors,, Multiple Linear Regression*** **Prediksi Banjir*****, Random Forest.***

## I. PENDAHULUAN

Bencana banjir merupakan sebuah fenomena alam yang disebabkan oleh banyak variabel di Indonesia, Wilayah Sungai (WS) adalah kesatuan wilayah pengelolaan sumber daya air dalam satu atau lebih daerah aliran sungai dan/atau pulau-pulau kecil yang luasnya kurang dari atau sama dengan 2.000 km2. Daerah Aliran Sungai (DAS) adalah suatu wilayah daratan yang merupakan satu kesatuan dengan sungai dan anak-anak sungai yang berfungsi menampung, menyimpan dan mengalirkan air yang berasal dari curah hujan ke danau atau ke laut secara alamiah yang batas di darat merupakan pemisah topografis dan batas di laut sampai dengan daerah perairan yang masih terpengaruh aktivitas daratan [1].

Banjir di Indonesia rata – rata disebabkan oleh luapan air dari sungai ke daratan, sungai memiliki banyak jenis dari mulai sungai besar maupun kecil. Menurut data yang dihimpun dalam RKPD, Kabupaten Tasikmalaya memiliki 4 (empat) Wilayah Sungai (WS) yang terbagi dalam 4 Daerah Aliran Sungai yaitu, DAS Citanduy dengan luas 10.695,19 Km2 dengan melintasi Kabupaten Tasikmalaya, Kota Tasikmalaya, Kabupaten Ciamis, Kota Banjar Provinsi Jawa Barat dan Kabupaten Cilacap Provinsi Jawa Tengah dan berhulu di Desa Guranteng Kecamatan Pagerageung Kab. Tasikmalaya, DAS Ciwulan dengan luas 236,6 Km2 merupakan sungai terbesar yang membelah Kabupaten Tasikmalaya dan berhulu di Gunung Karacak, Galunggung, Bungbulang, dan Balitiganar, rata-rata debit harian 2,37 – 2,65m3/detik, DAS Cimedang merupakan sungai yang terletak antara Perbatasan Kabupaten Tasikmalaya dan Kabupaten Ciamis dengan debit maksimum sebesar 89,44 m3/detik dan debit minimum 0,82 m3/detik., DAS Cilangla yang berhulu di Sukahurip rata-rata debit harian 1,77 – 23,6 m3/detik [2].

Kondisi Sungai di Kabupaten Tasikmalaya memiliki permasalahan yang beragam yang disebabkan oleh faktor alam dan faktor manusia. Bencana banjir seringkali terjadi di beberapa titik di Kabupaten Tasikmalaya salah satunya di Kecamatan Sukaresik, wilayah yang merupakan termasuk DAS Citanduy tersebut sudah dilanda banjir sejak tahun 2013 dan sampai saat ini masih menjadi langganan banjir setiap tahunnya dengan frekuensi banjir bisa sampai 3 kali dalam satu tahun. Pada 24 Juli 2013 terdapat lima kecamatan yang diterjang banjir, yakni Kecamatan Pancatengah, Sukaresik, Pagerageung, Ciawi, dan Rajapolah dengan ketinggian air dalam rumah sekitar 50-250 cm [3].

Penyebab banjir disebabkan beberapa faktor dalam hal ini yaitu hujan dengan intensitas tinggi, penanggulangan tidak optimal, luapan sungai citanduy, kerusakan lingkungan dan tidak ada pompa yang berfungsi dengan baik. Banjir yang terjadi menyebabkan beberapa kerugian antara lain ratusan rumah terendam, puluhan hektare sawah terendam, kerugian moril & material, berdampak pada kesehatan masyarakat dan kerusakan bangunan & infrastruktur. Kerugian terjadi karena tidak terdapat sistem pengukur yang lengkap terhadap variabel penyebab terjadinya banjir, tidak terdapat sistem deteksi dini sehingga tidak ada yang dapat dijadikan acuan untuk antisipasi oleh masyarakat dan penyebab lainnya dari faktor alam dan faktor manusia. Selain itu dilakukan juga Observasi dan wawancara kepada BBWS (Balai Besar Wilayah Sungai) Citanduy dalam hal ini sebagai pengelola dari aliran sungai citanduy tersebut, dari keterangan yang didapatkan Penelitian terakhir dari Litbang tahun 2016 sekitar 1 juta m3/tahun sedimentasi yang terjadi adalah ¾ berasal dari Sungai Citanduy. Hal tersebut mengakibatkan banjir di Kecamatan Sukaresik, sehingga terbentuknya kampung laut. Untuk itu perlu adanya rekomendasi secara continue agar kegiatan Pengendalian Daya Rusak Air bisa berlanjut.

Dari ringkasan permasalahan yang diuraikan, penulis menemukan gap sistem yang dapat dikembangkan khususnya dengan memperhatikan objek penelitian, data yang tersedia dan

model yang tepat untuk digunakan, maka perlu dilakukan *Penyusunan Model Untuk Predkisi Bencana Banjir Menggunakan Machine Learning*, Tujuan dari penelitian ini adalah Mengetahui *variabel* dan Algoritma dengan parameter yang tepat untuk melakukan prediksi Bencana Banjir khususnya di Kecamatan Sukaresik Kabupaten Tasikmalaya untuk selanjutnya dijadikan sebagai Rekomendasi Model atau Algoritma *Machine Learning* yang terbaik dalam melakukan Prediksi Bencana Banjir di wilayah tersebut.

Dalam penelitian ini data yang digunakan adalah data *real* yang didapatkan dari lokasi Penelitian di DAS Citanduy Wilayah sungai Desa Tanjungsari Kecamatan Sukaresik Kabupaten Tasikmalaya & Sumber data berasal dari Balai Besar Wilayah Sungai (BBWS) Citanduy dengan data yang tersedia terbatas yaitu hanya data Curah Hujan dan Debit air selama 3 Tahun terakhir 2019 – 2021.

## I. METODE

### *A. Jenis, Sifat dan Pendekatan Penelitian*

Metode penelitian yang digunakan dalam penelitian ini adalah metode eksperimen, yaitu metode yang bertujuan untuk menguji pengaruh suatu variabel terhadap variabel lain atau menguji bagaimana hubungan sebab akibat antara variabel yang satu dengan variabel yang lainnya. Metode penelitian eksperimen memiliki perbedaan yang jelas dibanding dengan metode penelitian lainnya, yaitu adanya pengontrolan terhadap variabel penelitian dan adanya pemberian perlakuan terhadap kelompok eksperimen[4].

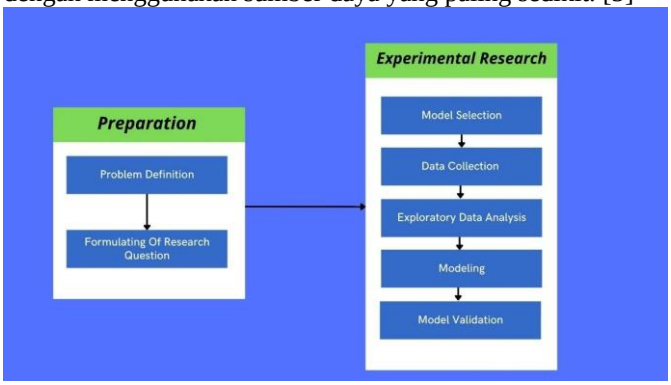
Sifat penelitian adalah deskriptif karena proses penelitian yang dilakukan akan menggambarkan bagaimana proses sebuah pola terjadinya banjir dan diuji akurasinya. Model yang dipilih dalam penelitian ini akan menghitung dan memprediksi terjadinya banjir berdasarkan data yang didapatkan di lapangan.

Pendekatan Penelitian Menggunakan pendekatan kuantitatif, karena variable diukur dengan indikator yang pasti dan bernilai.

### *B. Design an Experiment*

*Design an experiment* merupakan bagian integral dari proses eksperimen. Eksperimen adalah proses berulang dan multi-langkah yang terdiri dari aktivitas-aktivitas konkrit. Setiap langkah dalam proses tergantung pada keputusan yang dibuat pada langkah sebelumnya. Seringkali peneliti harus kembali ke langkah sebelumnya untuk mempertimbangkan kembali pilihan yang dibuat karena temuan pada langkah selanjutnya. Dengan cara ini, eksperimen adalah proses umpan-maju dengan loop umpan balik di setiap Langkah.

Desain eksperimental dimulai dengan pertanyaan penelitian. Jelas menyatakan pertanyaan penelitian sangat penting, karena semua keputusan selanjutnya mengalir dari pertanyaan penelitian. Dengan cara ini, kualitas desain eksperimen juga tergantung pada pertanyaan penelitian yang jelas. Pernyataan pertanyaan penelitian yang tidak jelas dapat menyebabkan desain eksperimen yang buruk, yang pada gilirannya mengakibatkan kegagalan keseluruhan penelitian. Tidak peduli seberapa kompleks analisis statistik dari data yang dihasilkan, dimulai dengan desain yang buruk akan mengarah pada kesimpulan yang lemah atau tidak memadai. Kualitas desain diukur dengan seberapa baik tujuan studi ditangani. Desain eksperimen yang baik akan menjawab pertanyaan kunci penelitian dengan menggunakan sumber daya yang paling sedikit. [5]




Gambar 1. Gambaran Skema dari proses eksperimental

### *C. Preparation*

*Preparation* dibagi menjadi dua fase, fase pertama adalah *Problem definition* atau pendefinisian masalah adalah proses awal yang dilakukan dalam penelitian ini, dalam mendefinisikan permasalahan yang terjadi dilapangan dilaksanakan beberapa tahapan berupa observasi dan wawancara dengan masyarakat khususnya di lokasi objek penelitian yaitu di Desa Tanjungsari Kecamatan Sukaresik Kabupaten Tasikmalaya. Dari hasil pendefinisian masalah didapatkan *Research Of Question* yang akan dijawab dengan menggunakan metode *experimental Research.*

### *D. Experimental Research*

***Model Selection***

Langkah pertama dalam *experimental Research* adalah *model selection*, Langkah ini dilakukan dengan melaksanakan Studi Literatur pada penelitian sebelumnya dengan tujuan mengumpulkan informasi terkait dengan Algoritma / Model prediksi yang pernah digunakan untuk memprediksi banjir.

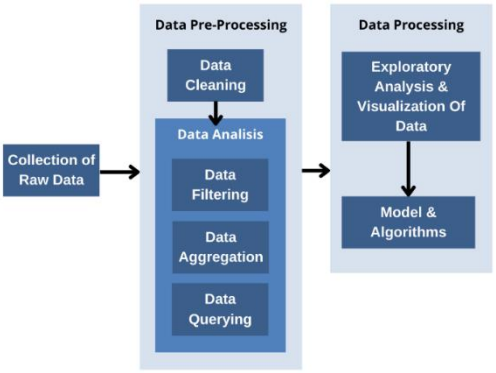
Dari beberapa penelitian dipilih Algoritma / Model terbaik untuk digunakan dalam penelitian ini. Pemilihan Model atau Algoritma untuk sistem prediksi banjir di Desa Sukaresik Kecamatan Tanjungsari dilakukan dengan metode *Research Gap* dari hasil *Literature Review* penerapan Teknologi AI pada sistem Deteksi Banjir. Pencarian *Research Gap* salah satunya dilakukan dengan mencari Melalui *Platform Open Knowledge Maps* & Melakukan analisis *bibliometric* menggunakan *vosviewer Publish Or Perish* untuk mengetahui Penelitian yang pernah dilakukan.

***Data Collection & Exploratory Data Analysis***

Terdapat beberapa metode yang digunakan baik dalam pengumpulan data maupun informasi yang diperlukan dalam penelitian ini, dalam fase ini khusus untuk mengumpulkan data real di lapangan terkait variabel yang telah ditentukan untuk merumuskan sebuah model *machine learning*. Data diperoleh dari beberapa instansi terkait yaitu Desa Tanjungsari Kecamatan Sukaresik Kabupaten Tasikmalaya mengenai histori banjir di wilayah tersebut dan data lainnya diperoleh dari Balai Besar Wilayah Sungai Citanduy (BBWS Citanduy) Jl. Prof.Ir.Sutami No.1, Karangpanimbal, Kec. Purwaharja, Kota Banjar.

Setelah mendapatkan data yang diperlukan selanjutnya dilakukan analisis data, dimana teknik analisis data ini sangat diperlukan agar tujuan penelitian tercapai dan teknik yang digunakan tergantung pada faktor utama jenis data itu sendiri,

berikut beberapa tahapan terkait dengan analisis data [6]
.




Gambar 2. Tahapan Analisis Data

***Modeling***

Dalam penelitian ini digunakan 5 model algoritma *machine learning* antara lain *Artificial Neural Network, Random Forest, Decision Tree, Multiple Linear Regression dan K Nearest Neighbors, modelling* dilakukan menggunakan layanan *google collabs*, lalu hasil dari proses pembuatan tersebut akan menciptakan model di mana diharapkan model tersebut dapat menghasilkan nilai peramalan terjadinya banjir yang paling optimal.

***Model Validation***

*Model Validation* menggunakan *K Fold Validation*, proses ini dilakukan dengan tujuan melakukan data *training* pada setiap model yang digunakan. Proses validasi dilakukan untuk mengukur akurasi dari *data training* [7].

K *fold Validation* adalah salah satu metode *cross validation* yang digunakan untuk meminimalisir terjadinya overlapping data testing, *Indeks K pada Fold Validation* adalah jumlah lipatan atau pembagian data testing yang dilakukan.

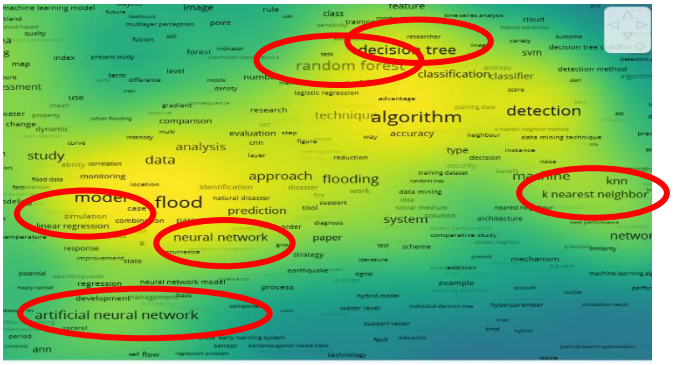
Selain itu digunakan juga *Confusion Matrix* untuk melihat hasil dari akurasi, presisi dan *recall* pada setiap variasi model yang digunakan [8].

## II. HASIL DAN PEMBAHASAN

Pada bagian ini berisi tentang proses penyusunan model algoritma *machine learning* dalam menghasilkan *output* atau rekomendasi terbaik untuk prediksi banjir di DAS Citanduy Desa Tanjungsari Kecamatan Sukaresik Kabupaten Tasikmalaya dengan menggunakan 5 Algoritma terpilih antara lain *Artificial Neural Network, Random Forest, Decision Tree, K Nearest Neighbors* dan *Multiple linear Regression* dimana proses dari perbandingan Algoritma tersebut dimulai dengan *Model Selection, Data Collection & Analysis, Design Of Experiment, Experiment* hingga pengujian model menggunakan secara manual dengan beberapa metode dan menggunakan *K Fold Cross Validation* dan *Confusion Matrix*.

### *A. Model Selection*

Tahapan awal persiapan dalam melakukan studi dan analisis machine learning untuk prediksi banjir adalah *Model Selection*. *Model Selection* dilakukan menggunakan analisis *bibliometric*, Analisis *bibliometric* adalah pemetaan *trend* riset penelitian dengan pengolahan metadata dari *google scholar* Tujuannya untuk mengetahui *trend* riset tentang Prediksi banjir khususnya menggunakan Algoritma *Machine Learning*. Pengambilan metadata menggunakan aplikasi *Publish or Perish (POP) versi 7.31* untuk selanjutnya dilakukan analisis terkait dengan keterhubungan antara topik dengan *keywords* yang digunakan.




Gambar 3. Visualisasi Vos Viewer

Hasil dari analisis *bibliometric* yang dilakukan menunjukan bahwa algoritma popular untuk penelitian mengenai prediksi banjir terdiri dari.

1. Artificial Neural Network
2. Random Forest
3. K Nearest Neighbors
4. Decision Tree
5. Linear Regression
6. Naïve Bayes
7. SVM
8. Decision Tree C.45
9. Convolutional Neural Network

Pada penelitian ini diambil 5 teratas dengan pengembangan khususnya untuk *Linear Regression* diganti menjadi *Multiple linear regression* hal ini dikarenakan disesuaikan dengan variabel yang tersedia di lapangan.

### *B. Data Collection & Exploratory Data Analysis*

*Data collection* dilaksanakan dengan mengumpulkan data terkait variabel terjadinya banjir dan diperoleh data selama 3 Tahun terakhir dengan jumlah data 1096. Dengan *sampel* data yang dapat dilihat pada tabel 1.

TABEL I
SAMPLE DATASET

| No | Curah Hujan | Debit | Banjir |
|---|---|---|---|
| 1 | 3.5 | 47.71 | Tidak Banjir |
| 2 | 60 | 63.67 | Banjir |
| 3 | 23.5 | 56.18 | Tidak Banjir |
| 4 | 1.5. | 64.41 | Tidak Banjir |
| 5 | 54 | 55.98 | Banjir |

Pengumpulan data dilakukan dengan meminta data secara langsung kepada Balai Besar Wilayah Sungai Citanduy (BBWS) dengan melakukan wawancara terkait data yang tersedia dan relevansi terkait terjadinya banjir di Desa Tanjungsari Kecamatan Sukaresik Kabupaten Tasikmalaya. Hasil dari pengumpulan data tersebut didapatkan dua variabel data yang tersedia yaitu data Curah hujan didapatkan dari pos curah hujan terdekat dengan lokasi objek penelitian dan data debit air yang didapatkan dari aliran sungai citanduy –

cirahong yang merupakan aliran sungai terusan dari objek penelitian. Sedangkan kondisi terjadinya banjir didapatkan dari hasil wawancara langsung dengan warga / Masyarakat desa Tanjungsari Kecamatan Tanjungsari Kabupaten Tasikmalaya.

Data yang digunakan pada penelitian ini adalah data dimulai tahun 2019 – 2021, untuk tahun selanjutnya label dapat berubah dengan nilai variabel yang didapatkan saat ini karena ada potensi perubahan struktur sungai yang terjadi baik perubahan secara alami berupa tambahan atau pengurangan sedimen dan luas sungai atau dengan adanya perubahan yang dilakukan oleh manusia.

Data Analisis menggunakan metode *EDA (Exploratory data Analysis)* hal ini digunakan untuk mendapatkan informasi atau *insight* dasar menggunakan visualisasi dan nilai – nilai dasar statistic sederhana. Dalam EDA dilakukan data *preparation* (Menangani dan mentransformasi data, *Noise*, *Missing Value*, Duplikasi dan *Outlier*). Dengan EDA (*Exploratory data Analysis*) dapat dilihat apakah perlu dilakukan transformasi pada data sebelum dilakukan pemodelan pada tahapan selanjutnya.

Langkah pertama dalam proses ini adalah menyediakan *packages* yang dibutuhkan sebagai berikut :

```
[102] import warnings; warnings.simplefilter('ignore')
      import scipy, itertools, pandas as pd, matplotlib.pyplot as plt, seaborn as sns, numpy as
      import matplotlib.cm as cm
      from collections import Counter
      from scipy import stats
      from sklearn.preprocessing import StandardScaler, MinMaxScaler

      plt.style.use('bmh'); sns.set()
```

Gambar 4. Package EDA

Selanjutnya dilakukan proses pemeriksaan tipe data, untuk memeriksa bahwa tipe data yang digunakan sesuai dengan kebutuhan model yang digunakan. Pada proses *modelling* data yang diharapkan adalah menggunakan *type data Float* untuk variabel Curah Hujan & Debit Air, dan *type* data Integer untuk Data Banjir pada data banjir digunakan angka 1 untuk mewakili terjadinya banjir & angka 0 untuk mewakili tidak terjadi banjir. Hasil dari proses ini adalah sebagai berikut :

```
banjir.info() #Memmeriksa Type data

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1096 entries, 0 to 1095
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Curah Hujan  1096 non-null   float64
 1   Debit        1096 non-null   float64
 2   Banjir       1096 non-null   int64
dtypes: float64(2), int64(1)
memory usage: 25.8 KB
```

Gambar 5. Pemeriksaan Tipe Data

Dalam gambar diatas dapat dilihat bahwa tipe data yang ada pada data set sudah sesuai dengan yang dibutuhkan pada saat proses *modelling*.

Tahapan selanjutnya dari Proses *EDA (Exploratory Data Analysis)* adalah memeriksa dan menghapus duplikasi data. Duplikasi data akan sangat mempengaruhi hasil dari proses *modelling* yang akan dilaksanakan selanjutnya, Hasil dari proses pemeriksaan duplikasi data didapatkan bahwa dari 1096 data terdapat duplikasi sebanyak 119 data, hal ini ditangani dengan menghapus duplikasi data dan data yang tersedia selanjutnya untuk dilakukan pemodelan adalah sebanyak 977 data.

```
[12] banjir.drop_duplicates(inplace=True)
     print(banjir.duplicated().sum())
     print (banjir.shape)

     0
     (977, 3)
```

Gambar 6. Menghilangkan Duplikasi Data

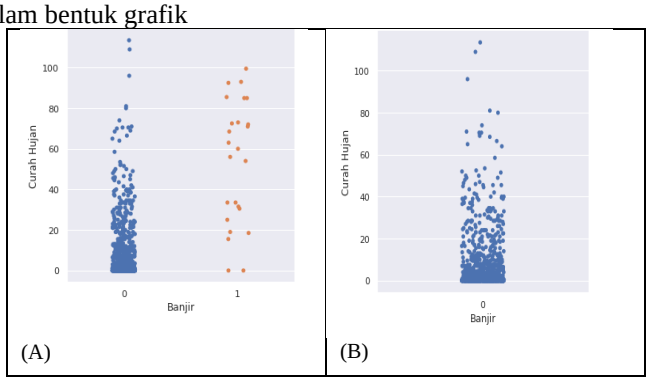
Tahapan selanjutnya dalam proses EDA (*Exploratory Data Analysis*) adalah pemeriksaan *Noise* pada data. *Noise* pada data biasanya terjadi dari pengumpulan sumber data, sepertinya misalnya disebabkan pada perangkat *sensing* data dikarenakan kerusakan, kesalahan *input* atau *entry data*, Transmisi yang tidak sempurna sampai dengan Inkonsistensi dalam penamaan data. Untuk menemukan *Noise* perlu dilakukan analisis terlebih dahulu terhadap *Outlier* pada data, *Outlier* adalah data yang memiliki karakteristik secara signifikan berbeda dengan kebanyakan data yang tersedia, *Outlier* bukanlah *noise* dalam artian *outlier* adalah data yang valid, dalam *big data* biasanya *Outlier* biasanya ditemukan dan dalam jumlah yang signifikan.

Dalam proses *EDA*, sering ditemukan mengenai *outlier* dan tidak jarang *outlier* tersebut di *Exclude* padahal sering ditemukan *outlier* tersebut merupakan bagian penting dari dataset yang akan digunakan. Menghindari hal tersebut dalam penelitian ini *outlier* tidak di *exclude* melainkan diolah secara terpisah dengan menggunakan model statistik dengan menggunakan *machine learning*, hal ini dilakukan agar dapat dilihat *noise* yang ada pada data termasuk pada *noise* yang perlu di *exclude* atau dipertahankan karena merupakan bagian penting dari *dataset*, dalam melakukan hal ini dilakukan perhitungan *Confidence Interval* (Interval Kepercayaan) untuk mengekspresikan ketepatan perkiraan pengukuran, dalam penelitian ini diasumsikan interval kepercayaannya adalah 95%.

```
df = np.abs(banjir.Banjir - banjir.Banjir.mean())<=(2*banjir.Banjir.std())
print (df.shape)
df.head()

(977,)
0    True
1    True
2    True
3    True
4    True
Name: Banjir, dtype: bool

asump = banjir [df]
print (asump.shape,banjir.shape)

(952, 3) (977, 3)
```

Gambar 7. Memeriksa Outlier pada Data

Gambar diatas menjelaskan bahwa terdapat outlier dengan jumlah data sebanyak 25 data, untuk memastikan bahwa data tersebut *noise* atau *outlier* maka dilakukan perbandingan dalam bentuk grafik




Gambar 8. Memeriksa Outlier pada Data

Gambar diatas adalah perbandingan sebelum data diolah dengan interval kepercayaan dan setelah diolah menggunakan interval

kepercayaan. Dari grafik tersebut dilihat bahwa data yang dianggap noise pada poin (b) merupakan *outlier* yang merupakan bagian penting dari dataset yang digunakan untuk pemodelan data. Kesimpulan dari tahapan ini adalah dataset dipertahankan sehingga tidak ada pengurangan data yang dianggap *noise* atau *outlier* yang harus di *exclude*.

Tahapan selanjutnya adalah memeriksa *Missing Value* pada dataset, untuk memastikan *missing value* dilakukan beberapa tahapan dengan pemeriksaan dengan grafik dan pemeriksaan data secara manual. *dataset* yang digunakan tidak terdapat *missing value* sehingga dapat digunakan untuk proses selanjutnya dalam pemodelan.

### *C. Modeling Menggunakan Artificial Neural Network*

Pada tahap ini melakukan beberapa tuning parameter dengan menentukan input, hidden layer dan output dari ANN, Selain itu Agar didapatkan model yang terbaik diperlukan perubahan pada parameter yang ada. Proses yang dilakukan untuk mendapatkan variasi model terbaik untuk ANN dilakukan dengan tahapan sebagai berikut :

a. Fungsi Pelatihan (*training function*)
*Training function* yang digunakan pada penelitian ini yaitu, *Adam Optimizer* algoritma optimasi yang dapat digunakan sebagai ganti dari prosedur *classical stochastic gradient descent* untuk memperbarui bobot secara iteratif yang didasarkan pada *data training*. *Adam* dapat dikatakan merupakan kombinasi antara *RMSprop* dan *Stochastic Gradient Descent* dengan *momentum*.

b. Fungsi *Learnrate* (*learnrate function*)
Pencarian nilai *learnrate*, juga berpengaruh pada performa model. Nilai yang digunakan juga sama yaitu mulai dari 0.1 hingga 0.9 dan dengan interval 0.1.

c. Penentuan Hidden Layer
Jumlah *node* pada *hidden layer* ini dapat memberi pengaruh pada model *neural network*. Juga ada variasi *node* sebanyak n * 3 *node* [39]. Jumlah ini berdasarkan metode aturan praktis dalam menentukan jumlah *neuron* yang tepat dalam penggunaan di *hidden layer*. Isi dari peraturan praktis tersebut ialah "Jumlah *hidden neuron* sebaiknya berukuran 3 kali dari *input layer*, ditambah ukuran *output layer*". Dikarenakan *input layer* adalah 2, maka dimulai penghitungan *node* adalah 2. Jadi misal *node* yang digunakan adalah dari 2 Hidden layer selanjutnya dilakukan untuk validasi hingga 2*3=6.
Dalam penelitian ini model *Artificial Neural Network* terdiri dari input layer yang isinya adalah *neuron-neuron* data-data Curah hujan dan debit di masa lalu, *hidden layer* yang terdiri dari satu layer berfungsi aktivasi dan *output layer* yang berisikan target prediksi banjir. *Hidden layer* memiliki nilai yang bergerak mulai dari *neuron* 2 sampai pada *neuron* 6 [9].

d. K Fold Cross Validation
Validasi terakhir untuk memperoleh akurasi yang maksimal adalah menggunakan *K Fold Validation*, *K Fold validation* digunakan setelah dilakukan uji coba dengan masing – masing parameter dan hidden layer yang berbeda – beda. Secara umum, *K fold Validation* akan membandingkan *n* model dalam *cross validation ini*, dalam arti lain fungsi dari penggunaan metode *cross validation* adalah

- Untuk mengetahui performa dari suatu model algoritma dengan melakukan percobaan sebanyak *k* kali
- Untuk meningkatkan tingkat performansi dari model tersebut
- Untuk mengolah data set dengan kelas yang seimbang

Dalam penelitian ini, digunakan *K Fold Cross Validation* dengan nilai K sebanyak 3 Macam yaitu, K1 =3 , K2 = 5 K3 =10. Proses validasi mempergunakan bantuan *google collabs*. Dari hasil validasi ini akan diperoleh persentase keberhasilan atau kebenaran prediksi

e. Uji Performa

Dalam melakukan tahapan uji performa dilakukan pengkodean untuk mengetahui variabel atau parameter, *hidden layer*, K – Fold yang disetting pada uji performa model, Dalam penelitian ini terdapat 135 Kombinasi model ANN dari setiap *node*. Hasil dari uji performa ini diharapkan menghasilkan nilai prediksi terbesar sehingga dapat dijadikan rekomendasi penggunaan model untuk dibandingkan dengan algoritma lainnya. Kolom kode model berisikan huruf variabel, dan dijelaskan maksud dari kode tersebut pada kolom arti dan nilainya pada tabel berikut ini:

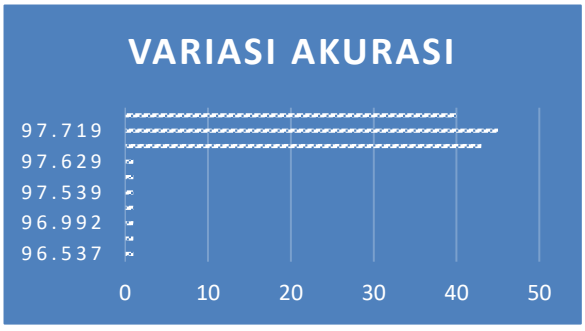
TABEL II
KOMBINASI MODEL ANN

| Kode Model | Artinya | Nilai |
|---|---|---|
| a | *Learning rate function* | 1=0.1… 9=0.9 |
| b | *Hidden Layer* | Sesuai *Hidden Layer* 2-6 |
| c | *K Fold* | Sesuai dengan *K Fold* 3, 5 atau 10 |

Kode model yang didapat bisa dibaca sesuai dengan penjelasan di Tabel diatas, Kode model yang didapatkan dari percobaan pertama adalah 1_2_3 Ini berarti model yang ditemukan adalah menggunakan *learn rate function* '0.1', *Hidden layer* '2' dan *K Fold Validation* '3'.

Selanjutnya disajikan table dari masing – masing *model test* yang dilakukan dengan menambah kolom hasil prediksi, hasil ini yang akan dibandingkan dengan algoritma lainnya untuk memberikan rekomendasi algoritma terbaik

Dari hasil validasi yang telah dilakukan dengan jumlah 135 kombinasi model menghasilkan variasi hasil sebanyak 10 variasi persentase akurasi prediksi




Gambar 9. Variasi Akurasi Model ANN

Dari gambar diatas dapat dijelaskan bahwa 135 kombinasi

menghasilkan variasi *persentase range* akurasi 96 – 97, dengan variasi tertinggi 97.719 dengan total hasil yang sama dari kombinasi model sebanyak 45 kombinasi.

Dari seluruh hasil percobaan yang dilakukan pada semua model serta perubahan parameternya, analisis dilakukan lebih lanjut pada model-model dengan nilai yang paling optimal. Dari hasil uji performa didapatkan bahwa *range* prediksi dengan beberapa variasi model yang dilakukan tidak terlalu memiliki perbedaan persentase hasil yang didapatkan. Tetapi yang Persentase prediksi terbaik dari hasil uji performa menggunakan *Artificial Neural Network* ini adalah dengan nilai Persentase 97,72%.

*D. Modelling Menggunakan Random Forest*

Agar didapatkan model yang terbaik diperlukan perubahan pada parameter yang ada. Pengubahan parameter yang ada pada penelitian ini dilakukan ini adalah sebagai berikut :

**a. Pembagian kombinasi *data training dan data testing***

Untuk memperoleh hasil prediksi terbaik dalam model *random forest*, pada penelitian ini dilakukan pembagian *data training* dan *data testing* antara lain dengan menggunakan kombinasi split 80%:20% .

**b. *Tuning Parameter* dengan *Grid Search CV***

Pada *Random Forest* terdapat nilai parameter yang diatur guna mendapatkan model yang optimal, yang disebut *hyperparameter*. *hyperparameter* digunakan untuk mengatur berbagai macam aspek dalam *machine learning* yang sangat berpengaruh pada performa dan model yang dihasilkan. Pencarian *hyperparameter* dilakukan secara manual atau dengan menguji kumpulan *hyperparameter* pada parameter yang ditentukan sebelumnya**.** Salah satu metode *hyperparameter* yang dapat diaplikasikan adalah *grid search*. *Grid search* merupakan metode alternatif yang digunakan untuk menemukan parameter terbaik dalam suatu model, sehingga metode yang digunakan secara akurat memprediksi data yang digunakan. *Grid search* dikategorikan sebagai metode yang teliti, karena dalam menentukan parameter terbaik dilakukan eksplorasi masing masing parameter dengan mengatur jenis nilai prediksi terlebih dahulu. Kemudian metode tersebut akan menampilkan skor untuk masing-masing nilai parameter.

Dalam penelitian ini Parameter yang digunakan untuk melakukan *hyperparameter* pada metode *Random Forest* sebagai berikut:

TABEL III
PARAMETER UJICOBA RANDOM FOREST

| **Parameter** | **Keterangan** |
|---|---|
| n_estimators | Jumlah pohon pada tree |
| min_samples_split | Pengukuran untuk kualitas split |
| min_samples_leaf | Jumlah sampel minimum yang dibutuhkan leaf node |
| max_features | Jumlah fitur yang dipertimbangkan saat mencari split terbaik |
| max_depth | Kedalaman maksimum pada tree |
| bootstrap | Metode pengambilan sampel titik data (dengan atau tanpa penggantian) |

Penelitian ini menggunakan 5 & 10 -fold cross validation yang digunakan untuk mengevaluasi kinerja model sebanyaklima kali & 10 kali perulangan dalam proses *grid search* dari setiap parameter. Nilai parameter terbaik dari proses *grid search* dengan fold 5 atau 10 digunakan dalam penentuan model klasifikasi.
Hasil dari *tuning parameter* menggunakan fungsi pada *scikit-learn* yaitu *gridsearchCV* disajikan dalam tabel berikut

TABEL IV
HASIL TUNING PARAMETER RANDOM FOREST

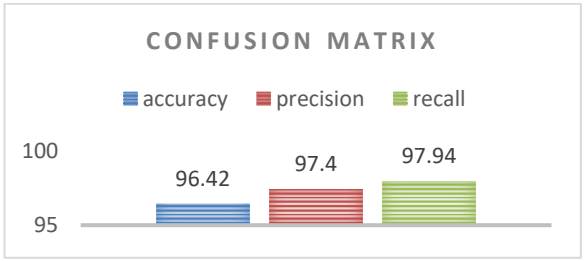
| **K – Fold** | Parameter | Grid Search Values | Best Parameter |
|---|---|---|---|
| **5** | n_estimators | 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000 | 600 |
| | min_samples_split | [2, 5, 10 | 2 |
| | min_samples_leaf | [1, 2, 4] | 4 |
| | max_features | auto', 'sqrt | auto |
| | max_depth | 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None | 70 |
| | bootstrap | True, False | True |
| **10** | n_estimators | 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000 | 1200 |
| | min_samples_split | [2, 5, 10] | 10 |
| | min_samples_leaf | [1, 2, 4] | 4 |
| | max_features | auto', 'sqrt | auto |
| | max_depth | 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None | 50 |
| | bootstrap | True, False | True |

Tabel diatas merupakan hasil tuning parameter yang didapatkan dari proses *gridsearchCV* dengan melakukan pencarian secara menyeluruh terhadap parameter yang diujikan. Dengan menggunakan 5-fold *cross validation* yang digunakan untuk mengevaluasi kinerja model sebanyak lima kali perulangan dalam proses *grid search* dari setiap parameter. Nilai parameter terbaik dari proses *grid search* digunakan dalam penentuan model klasifikasi. Pada klasifikasi *Random Forest* terdapat enam ratus pohon yang digunakan, kemudian setiap pohon membuat tujuh puluh percabangan. Akar kuadrat dari total fitur digunakan saat mencari split terbaik. Kemudian untuk mengukur kualitas *split* pada pohon digunakan nilai dua. Jumlah *sampel minimum* yang digunakan pada setiap *leaf node* sebanyak empat dengan *max_features auto* dan *bootstrap true*.

Sedangkan, dengan menggunakan menggunakan 10-fold *cross validation* yang digunakan untuk mengevaluasi kinerja model sebanyak sepuluh kali perulangan dalam proses *grid search* dari setiap parameter. Nilai parameter terbaik dari proses *grid search* digunakan dalam penentuan model klasifikasi. Pada klasifikasi *Random Forest* terdapat seribu dua ratus pohon yang digunakan, kemudian setiap pohon membuat lima puluh percabangan. Akar kuadrat dari total fitur digunakan saat

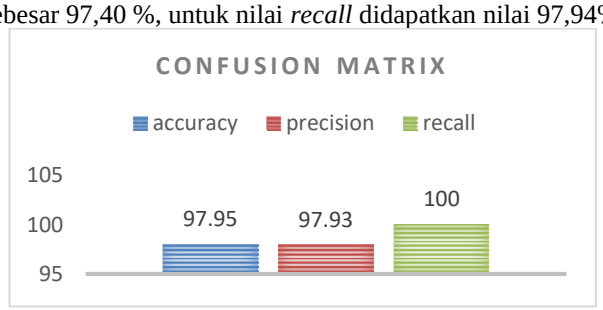
mencari *split* terbaik. Kemudian untuk mengukur kualitas *split* pada pohon digunakan nilai sepuluh. Jumlah sampel minimum yang digunakan pada setiap *leaf node* sebanyak empat dengan *max_features auto* dan *bootstrap true*.

Proses pengujian sistem dilakukan dengan metode *confusion matrix*. *Confusion matrix* dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari *True Positive* dan *True-Negative* memberikan informasi ketika classifier dalam melakukan klasifikasi data bernilai benar, sedangkan *False Positive* dan *False-Negative* memberikan informasi ketika classifier salah dalam melakukan klasifikasi data.




Gambar 10. Diagram Confusion Matriks Fold = 5

Gambar diatas adalah hasil perhitungan menggunakan confusion matrix untuk K Fold = 5 didapatkan nilai *accuracy* sebesar 96.42 %, untuk nilai *precision* didapatkan nilai sebesar 97,40 %, untuk nilai *recall* didapatkan nilai 97,94%.




Gambar 11. Diagram Confusion Matriks Fold = 10

Gambar diatas adalah hasil perhitungan menggunakan confusion matrix untuk K Fold = 10 didapatkan nilai *accuracy* sebesar 97.95 %, untuk nilai *precision* didapatkan nilai sebesar 97,93 %, untuk nilai *recall* didapatkan nilai 100%.

Dari hasil uji performa didapatkan model prediksi *random forest* memiliki Presantease Akurasi yang tinggi dengan menggunakan K Fold 10 dan parameter n_estimators=1200, max_depth=50, max_features=auto, min_samples_split=10, min_samples_leaf=4,bootstrap=True dengan nilai akurasi prediksi mencapai 97.95 %.

### *E. Modelling Menggunakan K – Nearest Neighbors*

Untuk memperoleh hasil prediksi terbaik dalam model *K Nearest Neighbors (KNN)* dilakukan *splitting* data *training* data *testing*, dalam model *KNN* ini variasi splitting data langsung menggunakan *splitting* K Fold *Validation*, dengan masing – masing KF1 = 5, KF2=7 dan KF3 =10. Dari masing – masing hasil tersebut akan dikombinasikan dengan variasi parameter untuk mendapatkan persentase akurasi terbaik.

*KNN* memiliki banyak parameter yang dapat dibuat untuk menjadi variasi, Prinsip kerja *K-Nearest Neighbor* (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan k tetangga (*neighbor*) terdekatnya dalam data pelatihan(training) . Dengan k merupakan banyaknya tetangga terdekat.

Dalam penelitian ini, karena k mempunyai peranan penting dalam training, maka untuk mendapatkan variasi model terbaik dilakukan pengubahan variasi pada Nilai K dimulai dari K1 – K7 dikombinasikan menggunakan *distance metric* Minkowski *distance* karena dianggap yang terbaik pada penelitian sebelumnya dan *metric* Minkowski dalam scikit learn sama dengan Euclidean *distance* = 2 [40], variasi tersebut selanjutnya dikolaborasikan dengan pengubahan *splitting* data menggunakan K Fold *Validation*.

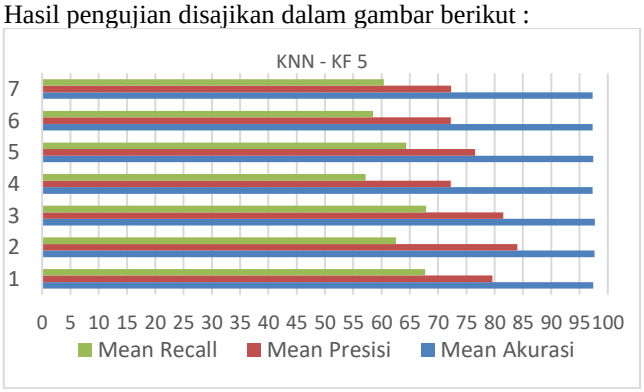
Tabel dibawah ini menggambarkan proses ujicoba model yang akan dilakukan validasi dengan menggunakan *Confusion matrix* untuk mengetahui akurasi , presisi dan *recall* terbaik.

TABEL V
PROSES UJI COBA MODEL KNN

| K – Fold Splitting | K (Tetangga) | Distance Metric |
|---|---|---|
| 5 | 1 | Minkowski distance |
| | 2 | Minkowski distance |
| | 3 | Minkowski distance |
| | … | Minkowski distance |
| | 7 | Minkowski distance |
| 7 | 1 | Minkowski distance |
| | 2 | Minkowski distance |
| | 3 | Minkowski distance |
| | … | Minkowski distance |
| | 7 | Minkowski distance |
| 10 | 1 | Minkowski distance |
| | 2 | Minkowski distance |
| | 3 | Minkowski distance |
| | … | Minkowski distance |
| | 7 | Minkowski distance |

Untuk mengetahui performa atau kinerja dari algoritma *K-Nearest Neighbor* dalam melakukan klasifikasi terhadap suatu kelas/label yang telah ditentukan, maka akan dilakukan pengujian pada hasil akurasi, presisi dan *recall*.

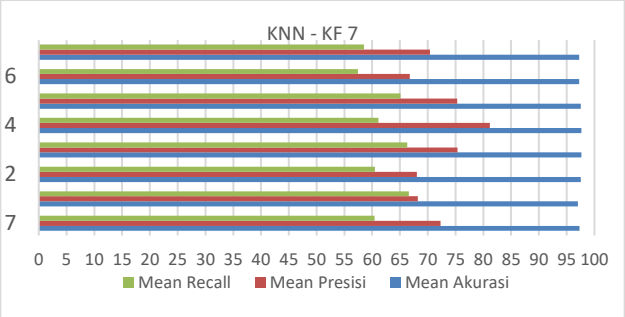
Hasil pengujian disajikan dalam gambar berikut :




Gambar 12. Diagram Hasil Variasi KNN K Fold = 5

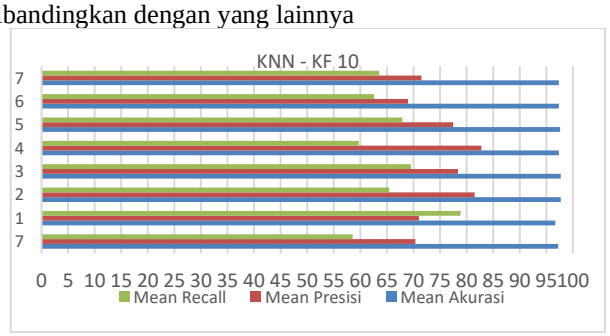
Hasil uji coba variasi model pada KNN dengan K Fold =5 memiliki *variative* yang beragam, nilai akurasi tertinggi pada variasi model ini adalah = 97,74% yang dimiliki oleh K1, K3, K5,

nilai *precision* tertinggi adalah 84,01% yang dimiliki oleh K2, sedangkan nilai *recall* tertinggi adalah 67,87% dimiliki oleh K3, kesimpulan dari variasi model KNN pada K Fold = 5 adalah K3 memiliki Performa terbaik karena memiliki akurasi, *recall* dan akurasi yang stabil dibandingkan dengan yang lainnya




Gambar 13. Diagram Hasil Variasi KNN K Fold = 7

Hasil uji coba variasi model pada KNN dengan K Fold =7 memiliki *variatif* yang beragam, nilai akurasi tertinggi pada variasi model ini adalah = 97,64% yang dimiliki oleh K3, K4, nilai *precision* tertinggi adalah 81,16% yang dimiliki oleh K4, sedangkan nilai *recall* tertinggi adalah 66,3% dimiliki oleh K3, kesimpulan dari variasi model KNN pada K Fold = 7 adalah K3 memiliki Performa terbaik karena memiliki akurasi, *recall* dan akurasi yang stabil dibandingkan dengan yang lainnya




Gambar 14. Diagram Hasil Variasi KNN K Fold = 10

Hasil uji coba variasi model pada KNN dengan K Fold =7 memiliki *variative* yang beragam, nilai akurasi tertinggi pada variasi model ini adalah = 97,75% yang dimiliki oleh K2, nilai *precision* tertinggi adalah 82,20% yang dimiliki oleh K4, sedangkan nilai *recall* tertinggi adalah 78,85% dimiliki oleh K1, kesimpulan dari variasi model KNN pada K Fold = 10 adalah K1 memiliki Performa terbaik karena memiliki akurasi, *recall* dan akurasi yang stabil dibandingkan dengan yang lainnya.

Dari hasil uji performa didapatkan dari variasi model KNN K=3 memiliki performa terbaik dengan memiliki keunggulan performa dengan K Fold = 5 & K Fold = 7, dari kedua K Fold tersebut persentase akurasi tertinggi dari KNN K=3 adalah 97,74% dengan Nilai *precision* 84,01% dan Nilai *Recall* 67,87%.

### *F. Modelling Menggunakan Multiple Linear Regression*

*Multiple linear regression* biasanya digunakan untuk meneliti hubungan antar dua atau lebih variabel, dengan paling tidak satu variabel sebagai variabel *dependen* (respon) dan variabel lainnya sebagai variabel independen (variabel prediktor), sehingga tidak memiliki banyak parameter yang dapat divariasikan.

Dalam penelitian ini, untuk menguji hasil terbaik dari prediksi *Multiple Linear Regression* digunakan *tuning parameter* pada *splitting* data. K Fold *Cross Validation* memiliki parameter tunggal yang disebut k yang mengacu pada jumlah grup yang akan dipecah menjadi sampel data tertentu.

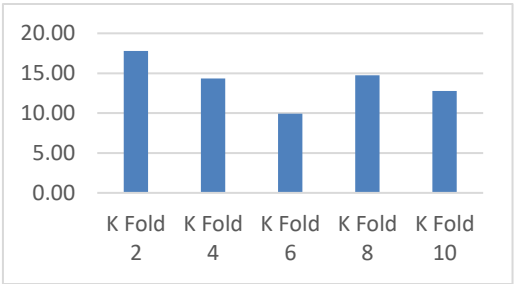
Dalam penelitian ini, digunakan K Fold *Cross Validation* dengan nilai K sebanyak 5 Macam yaitu, K1 =2 , K2 = 4 K3 = 6, K4 = 8, K5=10. Proses validasi mempergunakan bantuan. Dari hasil validasi ini akan diperoleh persentase keberhasilan atau kebenaran prediksi

Untuk mengetahui performa atau kinerja dari algoritma *Multiple Linear Regression* dalam melakukan prediksi terhadap suatu kelas/label yang telah ditentukan, maka akan dilakukan pengujian pada hasil akurasi.

Hasil pengujian disajikan dalam tabel berikut:

TABEL VI
HASIL UJI PERFORMA MLR

| K Fold *Splitting* | Hasil Akurasi |
|---|---|
| 2 | 17.81% |
| 4 | 14.35 |
| 6 | 9.92 |
| 8 | 14.75 |
| 10 | 12.79 |




Gambar 15. Diagram Hasil Variasi Model MLR

Dari hasil uji performa didapatkan dari variasi 5 K Fold , K = 2, K =4, K =6 , K =8 dan K=10, didapatkan hasil bahwa data dengan splitting K = 4 memiliki persentase tertinggi dengan hasil prediksi 17,81%.

### *G. Modelling Menggunakan Decision Tree*

Pada *Decision Tree* sama dengan *Random Forest* terdapat nilai parameter yang diatur guna mendapatkan model yang optimal, yang disebut *hyperparameter*. *hyperparameter* digunakan untuk mengatur berbagai macam aspek dalam *machine learning* yang sangat berpengaruh pada performa dan model yang dihasilkan. *Grid search* dikategorikan sebagai metode yang teliti, karena dalam menentukan parameter terbaik dilakukan eksplorasi masing masing parameter dengan mengatur jenis nilai prediksi terlebih dahulu. Kemudian metode tersebut akan menampilkan skor untuk masing-masing nilai parameter.

Dalam penelitian ini Parameter yang digunakan untuk melakukan *hyperparameter* pada metode *Decision Tree* adalah sebagai berikut

TABEL VII
TUNING PARAMETER DECISION TREE

| Parameter | Keterangan |
|---|---|
| min_samples_split | Pengukuran untuk kualitas split |
| min_samples_leaf | Jumlah sampel minimum yang dibutuhkan leaf node |
| max_depth | Kedalaman maksimum pada tree |
| criterion | fungsi yang digunakan untuk mengukur kualitas pemisahan |

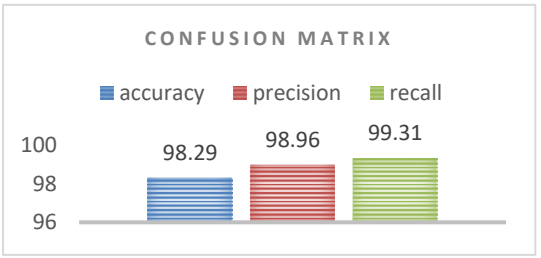
Penelitian ini menggunakan 3, 5, 7 -fold cross validation yang digunakan untuk mengevaluasi kinerja model sebanyak 3 kali, 5 Kali & 7 kali perulangan dalam proses grid search dari setiap parameter. Nilai parameter terbaik dari proses *grid search* dengan fold 3,5, fold 7 digunakan dalam penentuan model klasifikasi.
Hasil dari *tuning parameter* menggunakan fungsi pada *scikit-learn* yaitu *gridsearchCV* disajikan dalam tabel berikut :

TABEL VIII
PARAMETER TERBAIK DECISION TREE

| K – Fold | Total Kombinasi Model | Parameter | *Grid Search Values* | *Best Parameter* |
|---|---|---|---|---|
| 5 | 540 | min_samples_split | [2, 3, 4] | 2 |
| | | min_samples_leaf | [1, 2, 3] | 1 |
| | | max_depth | [2,4,6,8,10,12] | 2 |
| | | criterion | ['gini', 'entropy'], | 'entropy' |
| 7 | 756 | min_samples_split | [2, 3, 4] | 2 |
| | | min_samples_leaf | [1, 2, 3] | 2 |
| | | max_depth | [2,4,6,8,10,12] | 8 |
| | | criterion | ['gini', 'entropy'], | entropy |
| 3 | 324 | min_samples_split | [2, 3, 4] | 2 |
| | | min_samples_leaf | [1, 2, 3] | 1 |
| | | max_depth | [2,4,6,8,10,12] | 2 |
| | | criterion | ['gini', 'entropy'], | gini |

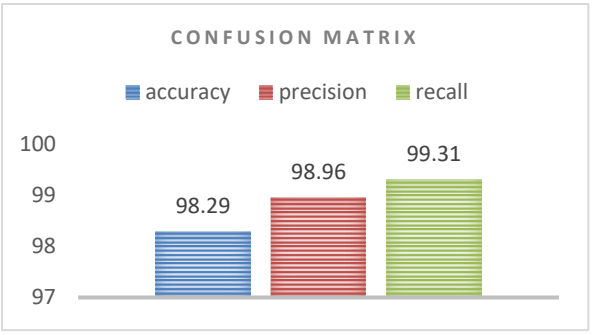
Tabel diatas merupakan hasil *tuning parameter* yang didapatkan dari proses *gridsearchCV* dengan melakukan pencarian secara menyeluruh terhadap parameter yang diujikan. Dengan menggunakan 3,.5.7 -fold *cross validation* yang digunakan untuk mengevaluasi kinerja model dengan total kombinasi model sebanyak 1620. Nilai parameter terbaik dari proses *grid search* digunakan dalam penentuan model klasifikasi.

Selanjutnya Kemudian dilakukan evaluasi model dengan melakukan prediksi terhadap data *testing* menggunakan *decision tree*. Dengan mengevaluasi model digunakan nilai akurasi, *precision*, dan *recall*



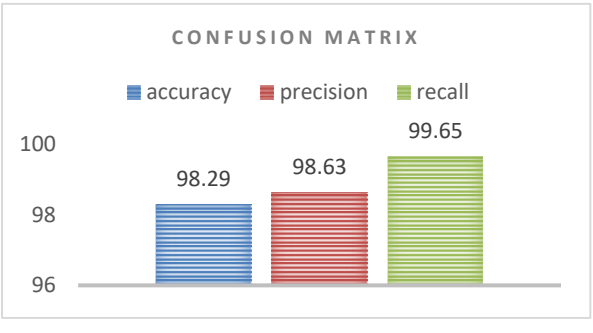

Gambar 15. Diagram Hasil Uji Performa Confusion Matriks K Fold = 3

Dari gambar diatas dapat dilihat bahwa didapatkan nilai *accuracy* sebesar 98.29 %, untuk nilai *precision* didapatkan nilai sebesar 98,96 %, untuk nilai *recall* didapatkan nilai 99,31%.




Gambar 16. Diagram Hasil Uji Performa Confusion Matriks K Fold = 5

Dari gambar diatas dapat dilihat bahwa didapatkan nilai *accuracy* sebesar 98.29 %, untuk nilai *precision* didapatkan nilai sebesar 98,96 %, untuk nilai *recall* didapatkan nilai 99,31%




Gambar 17. Diagram Hasil Uji Performa Confusion Matriks K Fold = 7

Dari gambar diatas dapat dilihat bahwa didapatkan nilai *accuracy* sebesar 98.29 %, untuk nilai *precision* didapatkan nilai sebesar 98,63 %, untuk nilai *recall* didapatkan nilai 99,65%
Dari hasil uji performa didapatkan model prediksi *Decision Tree* Rata - Rata memiliki Persentase Akurasi yang sama,

tetapi terdapat perbedaan pada tingkat *presisi* dan *recall*, maka dari itu hasil terbaik dengan menggunakan K Fold 7 dan
parameter max_depth=8, min_samples_split=2, min_samples_leaf=2.

## IV. SIMPULAN

Dari penelitian yang dilakukan, didapatkan hasil terbaik dari setiap algoritma yang digunakan dengan parameter yang bervariasi dan dilakukan uji validasi dengan hasil kombinasi terbaik disajikan pada tabel berikut ini.

TABEL IX
PERBANDINGAN HASIL AKURASI 5 ALGORITMA

| **Algoritma** | **Parameter** | **Hasil Akurasi** |
|---|---|---|
| ANN | Hidden Layer =4<br>Learning Rate = 9<br>K Fold = 10 | 97,72%. |
| Random Forest | n_estimators=1200,<br>max_depth=50,<br>max_features=auto,<br>min_samples_split=10,<br>min_samples_leaf=4,<br>bootstrap=True | 97.95 %. |
| KNN | K=3, Distance = Minkowski | 97,74% |
| Multiple Linear Regression | fit_intercept: bool = True, normalize: str = "deprecated", | 17.81% |
| Decision Tree | max_depth=8,<br>min_samples_split=2,<br>min_samples_leaf=2. | 98,29% |

Dalam perbandingan beberapa Algoritma dengan variasi tuning parameter , Algoritma terbaik dalam memprediksi banjir dengan dataset yang diambil dari Sungai Citanduy, Desa Tanjungsari Kecamatan Sukaresik adalah Algoritma Decision Tree dengan Persentase Hasil 98,29% dengan parameter max_dept=8, min_sample_split = 2, min_sample_leaf = 2.

## REFERENSI


[1] Kementerian Pekerjaan Umum dan Perumahan Rakyat, “Wilayah Sungai,” 2020. https://data.pu.go.id/dataset/wilayah-sungai (accessed Jan. 28, 2022).

[2] D. A. N. Kebijakan and K. Daerah, “Pemerintah kabupaten tasikmalaya,” pp. 1–50, 2017.

[3] R. Putra, . Z., E. Madona, and A. Nasution, “Desain dan Implementasi Peringatan Dini Banjir Menggunakan Data Mining dengan Wireless Sensor Network,” *J. Nas. Tek. Elektro*, vol. 5, no. 2, p. 181, 2016, doi: 10.25077/jnte.v5n2.261.2016.

[4] U. P. Indonesia, “Metodologi dan Strategi Penelitian,” *Pap. Knowl. . Towar. a Media Hist. Doc.*, p. 125, 2014.

[5] H. J. A. and G. J. Mellenbergh, *Research Methodology in the Social, Behavioural and Life Sciences*. 1999.

[6] S. Agarwal, *Data mining: Data mining concepts and techniques*. 2014. doi: 10.1109/ICMIRA.2013.45.

[7] M. Luckert and M. Schaffer-Kehnert, “Using Machine Learning Methods for Evaluating the Quality of Technical Documents,” *M.S. thesis, Dept. Comput. Sci. Linnaeus Unv., Sweden*, p. 102, 2015, [Online]. Available: http://www.diva-portal.org/smash/get/diva2:920202/FULLTEXT01.pdf

[8] D. Anguita *et al.*, “K – Fold Cross Validation for Error Rate Estimate in Support Vector Machines”.

[9] J. Heaton, *Neural networks editorial board addresses and specialties*, vol. 17, no. 1. 2004. doi: 10.1016/s0893-6080(03)00295-8.