BAB III METODOLOGI

A. Waktu dan Tempat Penelitian

Penelitian dimulai pada bulan Mei 2023 sampai dengan bulan Maret 2026 dan dilaksanakan di Stasiun Meteorologi Tunggul Wulung Cilacap yang berlokasi di Jl. Gatot Subroto No.20, Tambaksari, Sidanegara, Cilacap Tengah, Kabupaten Cilacap, Jawa Tengah 53223 atau berada di Balai Besar Wilayah II

B. Sumber dan Jenis Data

Sumber data dalam penelitian ini yaitu data sekunder, data sekunder adalah data tidak langsung dari sumber yang sudah ada dan dikumpulkan oleh pihak lain dalam buku, jurnal, atau laporan. Pada penelitian ini data diperoleh dari data *Synop ME48* yang telah di *entry* setelah observasi di lapangan pengamatan. Data yang digunakan pada penelitian ini merupakan data harian berupa data curah hujan harian, suhu udara rata-rata, kelembapan rata-rata, takanan udara rata-rata, kecepatan angin rata-rata, dan lama penyinaran matahari. Jenis data pada penelitian ini yaitu kuantitatif dimana data berupa angka atau numerik.

C. Variabel Penelitian

Variabel dalam penelitian terdiri atas dua jenis, yaitu variabel independen dan variabel dependen. Variabel independen merupakan variabel yang berperan dalam mempengaruhi akau menyebabkan terjadinya perubahan pada variabel lain. Variabel ini umumnya dapat dikendalikan oleh peneliti dalam proses penelitian. Sementara itu, variabel dependen adalah variabel yang dipengaruhi oleh variabel indepeneen, sehingga perubahan yang terjadi pada variabel ini merupakan akibat dari perubahan pada variabel independen.

Pemilihan variabel independen dan dependen pada penelitian ini didasarkan pada temuan dari penelitian sebelumnya. Salah satu penelitian yang menjadi rujukan adalah studi yang dilakukan oleh Deden Martia Nanda, Tacbir Hendro Pudjiantoro dan Pudpita Nurul pada tahun 2022 [6]. Dalam penelitian tersebut, variabel independen yang digunakan meliputi temperatue/suhu udara rata-rata, kelembapan rata-rata, kecepatan angin rata-rata dan curah hujan, sedangkan variabel dependen adalah kategori curah hujan [6]. Penelitian tersebut memanfaatkan data dengan rentang waktu selama enam tahun. Selain itu, penelitian yang dilakukan oleh Mohammad Yusuf R. R., Muhammad Valensyah A., dan Wawan Gunawan pada tahun 2021 juga dijadikan acuan [14]. Penelitian tersebut menggunakan beberapa variabel independen, yaitu kelembapan, suhu, suhu minimum, suhu maksimum, kecepatan rata-rata, kecepatan maksimum dan curah hujan, dengan variabel dependennya adalah suhu dan cuaca [14]. Penelitian tersebut menggunakan data dengan periode pengamatan selama sembilan bulan.

Sementara itu, periode rentang aktu yang digunakan dalam penelitian ini mencakup rentang aktu selama lima tahun, taitu dari tahun 2021 samapi tahun 2025. Pemilihan periode tersebut didasarkan pada rujukan dari penelitian-penelitian sebelumnya, serta pertimbangan bahwa data dengan rentang waktu antara lima sampai sepuluh tahun kerap digunakan dalam pengembangan model prediksi harian berbasis metode statistika ataupun *machine learning*.

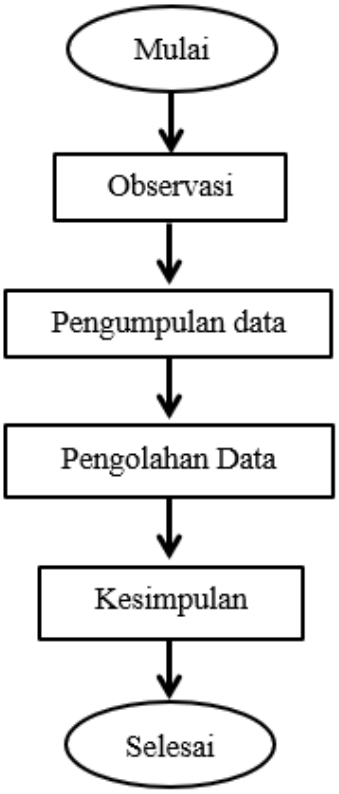
Dengan demikian, penelitian ini menggunakan variabel independen yang dipilih dengan mempertimbangkan berbagai faktor yang mempengaruhi curah hujan harian yang terdapat pada landasan teori serta merujuk pada penelitian-penelitian sebelumnya. Selain itu penelitian ini juga melakukan penambahan variabel guna memperbanyak informasi yang digunakan dalam proses analisis. Adapun variabel independen dan variabel dependen yang digunakan dalam penelitian ini, adalah sebagai berikut:

Tabel 1. Variabel Penelitian

Jenis Variabel	**Nama Variabel**	**Keterangan**
Variabel Independen (X)	Suhu Udara	Suhu udara rata-rata dalam satuan derajat Celcius (℃).
	Kelembaban	Kelembaban rata-rata dalam satuan Presentase (%).
	Tekanan Udara	Tekanan udara rata-rata dalam satuan Milibar (mb).
	Kecapatan Angin	Kecapatan angin rata-rata dalam satuan Knot.
	Lama Penyinaran Matahari	Lama penyinaran matahari dalam satuan waktu Jam.
Variabel Dependen (Y)	Curah Hujan Harian dengan Kategori:	Curah hujan harian dalam satuam Milimeter (mm) 1. Tidak Hujan 2. Hujan Ringan 3. Hujan Sedang 4. Hujan Lebat 5. Hujan Sangat Lebat 6. Hujan Ekstrem

D. Prosedur Penelitian

Prosedur penelitian yaitu langkah-langkah yang digunakan untuk mengumpulkan data guna menjawab permasalahan penelitian yang diajukan dalam penelitian ini. Adapun *flowchart* tahapan penelitian yang akan dilakukan oleh peneliti, yaitu sebagai berikut:



Gambar 1. *Flowchart* Tahapan Penelitian

Langkah-langkah yang akan dilakukan dalam penelitian ini adalah sebagai berikut:

1. Observasi
   Observasi dilakukan peneliti untuk mencari informasi serta mendapatkan data untuk penelitian.
2. Pengumpulan Data
   Data yang digunakan dalam penelitian ini merupakan data sekunder. Data yang diambil ialah data yang diperoleh dari pengamatan curah hujan harian, suhu udara rata-rata, kelembapan rata-rata, takanan udara rata-rata, kecepatan angin, dan lama penyinaran matahari. Data-data tersebut diambil tahun 2021 sampai dengan tahun 2025. Data yang telah diambil kemudian disusun secara terstruktur.
3. Pengolahan Data
   Pengolahan data dilakukan dengan menggunakan algoritma *K-Nearest Neighbor* (KNN).
4. Kesimpulan
   Kesimpulan diperoleh dari proses analisis data.

D. Analisis Data

Langkah-langkah dalam menganalisis data pada penelitian ini adalah sebagai berikut:

1. Dataset
   Pada dataset memuat beberapa data yang akan digunakan pada penelitian ini yaitu berupa data harian berdasarkan data curah hujan, suhu udara rata-rata, kelembapan rata-rata, takanan udara rata-rata, kecepatan angin rata-rata, dan lama penyinaran matahari yang diambil dari data SYNOP yang diambil dari bulan Januari 2021 sampai dengan bulan Desember 2025. Data tersebut dibagi menjadi dua, yaitu data *training* dan data *testing* dengan acuan 80:20 sebagai perbandingannya.
2. Kategori Curah Hujan
   Data curah hujan harian yang sudah tersedia kemudian diberikan kategori dengan ketentuan yang dapat dilihat pada tabel 2.
3. Normalisasi Data
   Data yang mempunyai rentang selisih cukup banyak dapat menghasilkan model dengan nilai akurasi yang kecil. Proses normalisasi data dapat dilakukan sebagai upaya untuk mengubah data yang mempunyai rentang cukup banyak. Normalisasi data sangat penting untuk algoritma *K-Nearest Neighbor* (KNN), karena algoritma tersebut mengandalkan jarak untuk menentukan tetangga terdekat. Normalisasi memastikan bahwa semua data diperlakukan setara dengan menskalakan nilainya ke rentang yang serupa, sehingga meningkatkan akurasi dan kinerja model secara keseluruhan.
   a) Data *Training*
      Data *training* (latih) merupakan data yang digunakan untuk melatih algoritma atau mambangun model. Data *training* diambil sebasar 80% dari keseluruhan data.
   b) Data *Testing*
      Data *testing* (uji) merupakan data yang akan dilakukan pengujian serta digunakan untuk pengambilan keputusan. Data *testing* diambil sebesar 20% dari keseluruhan data.
4. *K-Nearest Neighbor* (KNN)
   Pada proses ini model *K-Nearest Neighbor* (KNN) diaktifkan untuk menjalankan model, sehingga akan menampilkan hasil prediksi serta akurasi baik dari data *training* maupun data *testing*.
   a) Hasil Prediksi dari Data *Training*
      Pada proses ini akan diperoleh hasil prediksi dari data *training*.
      Akurasi dari data *training* dihitung dengan membandingkan nilai aktuan dari data *training* dengan nilai prediksi dari data *training*.
   b) Hasil Prediksi dari Data *Testing*

Pada proses ini akan diperoleh hasil prediksi dari data *testing*.

Akurasi dari data *testing* dihitung dengan membandingkan nilai aktuan dari data *testing* dengan nilai prediksi dari data *testing*.

5. Evaluasi Hasil

Evaluasi hasil diperoleh dari 2 (dua) teknik, yaitu sebagai berikut:

a) *Confussion Matrix*

Confussion matrix ialah melakukan pengujian untuk memperkirakan objek yang benar dan salah. *Confussion matrix* banyak digunakan dalam penelitian untuk mengevaluasi hasil dan mengukur kinerja suatu metode klasifikasi dan juga digunakan untuk menghitung serta menarik kesimpulan dari hasil penelitian. *Confussion matrix* dapat dilihat pada Tabel 9 dibawah ini [28]:

Tabel 2. *Confussion Matrix*

		Nilai Aktual	
		True	**False**
Nilai Prediksi	**True**	TP *(True Positive)* *Correct result*	FP *(False Positive)* *Unexpected result*
	False	FN *(False Negative)* *Missing result*	TN *(True Negative)* *Correct absence of result*

Keterangan:

TP : Banyaknya nilai prediksi yang sesuai dengan nilai aktual (bersifat positif).

FP : Banyaknya nilai prediksi yang tidak sesuai dengan nilai aktual (bersifat positif).

FN : Banyaknya nilai prediksi yang tidak sesuai dengan nilai aktual (bersifat negatif).

TN : Banyaknya nilai prediksi yang sama dengan nilai aktual (bersifat negatif).

Perhitungan performa kinerja dari klasifikasi dapat dihitung menggunakan persamaan berikut [28]:

1) *Accuracy*

   Accuracy merupakan proporsi jumlah prediksi benar. *Accuracy* digunakan untuk mengukur seberapa akurat algoritma dapat mengklasifikasikan data dengan benar. *Accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual. Berikut merupakan persamaan untuk menghitung *accuracy*:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%$$

2) *Recall*

   Recall merupakan proporsi kasus positif yang diidentifikasi dengan benar. *Recall* digunakan untuk menggambarkan keberhasilan algoritma dalam menemukan kembali sebuah informasi. Berikut persamaan untuk menghitung *recall*:

$$Recall = \frac{TP}{TP + FN} \times 100\%$$

3) *Precision*

   Precision merupakaan proporsi prediksi kasus positif yang benar. *Precision* digunakan untuk menggambarkan tingkat keakuratan antara data prediksi benar positif yang diminta dengan hasil prediksi yang diberikan algoritma. Berikut persamaan untuk menghitung *precision*:

$$Precision = \frac{TP}{FP + TP} \times 100\%$$

4) *F1 Score*

   F1 score merupakan perbandingan *precision* dan *recall* yang dibobotkan. Berikut persamaan *f – score*:

$$F - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\%$$

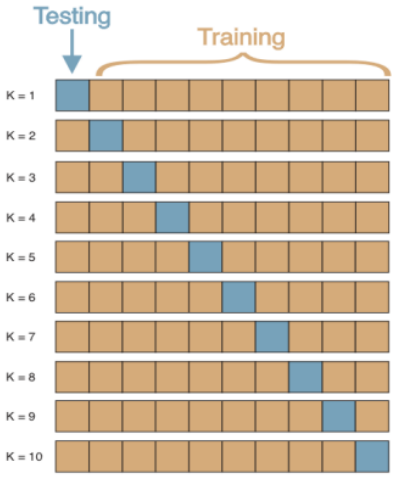
Berikut kriteria performa klasifikasi berdasarkan akurasi dapat dilihat pada Tabel 10 sebagai berikut [27]:

Tabel 3. Kriteria Performa Klasifikasi Berdasarkan Akurasi

Performa Klasifikasi	**Rentang Nilai**
Sangat Buruk	$\leq 60\%$
Buruk	$61\% - 70\%$
Cukup	$71\% - 80\%$
Baik	$81\% - 90\%$
Sangat Baik	$91\% - 100\%$

b) *Cross-Validation Score*

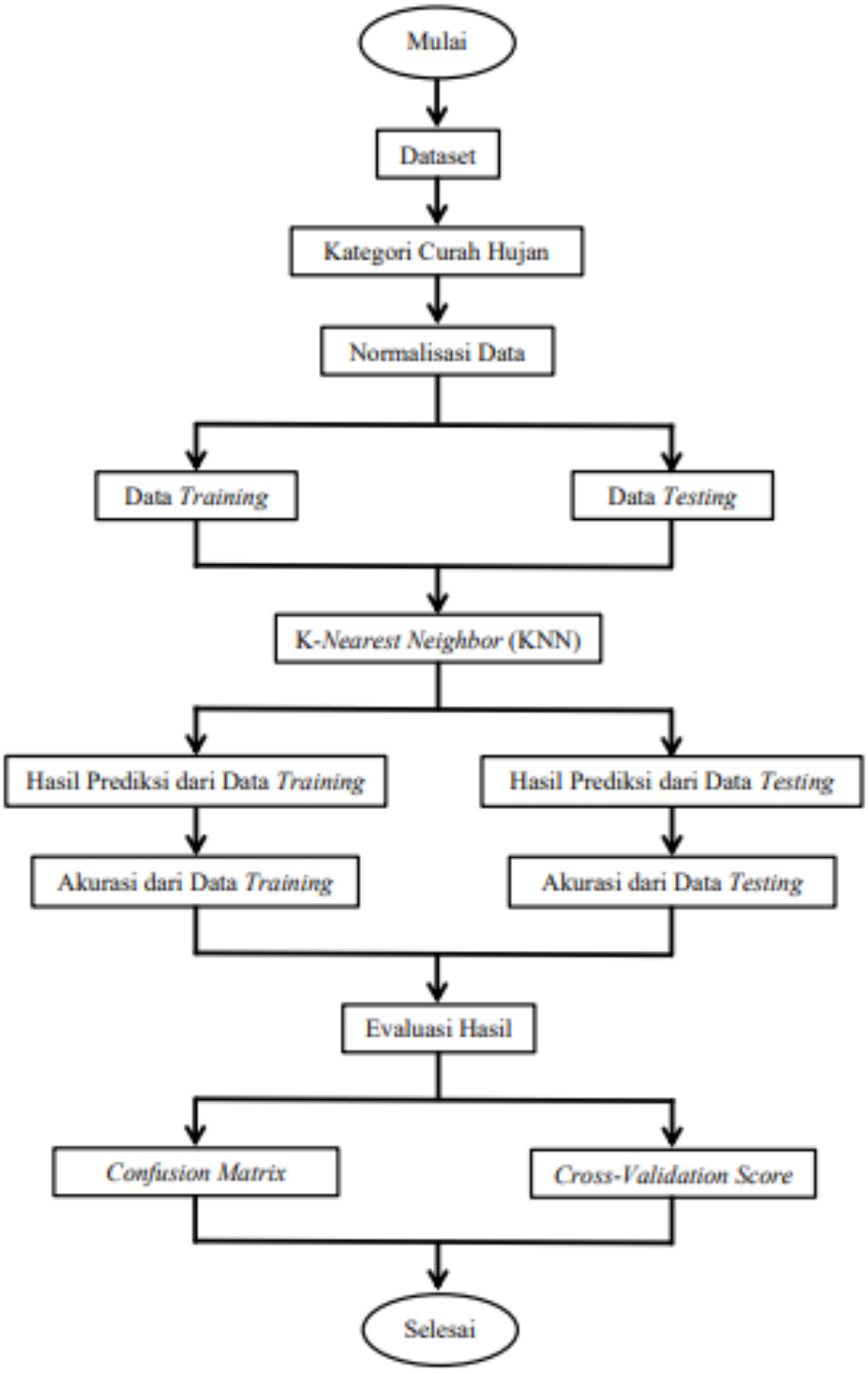
Cross Validation merupakan standar pemeriksaan yang dilakukan untuk memperkirakan kesalahan kinerja suatu algoritma atau model. *K-Fold Cross Validation* dilakukan pada proses klasifikasi data awal secara acak dibagi menjadi *k* subset berukuran sama sesuai kebutuhan. Pada *cross validation*, proses *training* dan *testing* dikerjakan sebanyak *k* kali. Pada penelitian ini, data akan dibagi menjadi *10-fold cross validation* yang sama rata. Kemudian data tesebut dieksekusi 10 kali untuk setiap subset data yang akan memiliki peluang untuk dijadikan sebagai data *testing* atau data *training*. Berikut merupakan gambaran model penggunaan *10-fold cross validation* [9]:



Gambar 2. *10-Fold Cross Validation*

Dengan menggunkan teknik *k-fold cross validation*, dapat memastikan bahwa perfoma model tidak hanya baik pada subset data tertentu tetapi juga pada seluruh dataset.

Berikut *flowchart* analisis data pada penelitian ini, yaitu sebagai berikut:



Gambar 3. *Flowchart* Analisis Data

Berikut *syntax* Python yang akan digunakan dalam algoritma K-*Nearest Neighbor* (KNN) [24]:

1) Import library yang diperlukan
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```
2) Import google drive
```
from google.colab import drive
drive.mount('/content/drive')
```
3) Menampilkan dataset
```
df = pd.read_excel("/content/drive/MyDrive/Uji Coba - Data Penelitian Tugas Akhir.xlsx")
df
```
4) Menampilkan informasi dasar dari dataset
```
df.drop('Tanggal', axis=1, inplace=True)
df.info()
```
5) Menghitung kemunculan kategori
```
df['Kategori'].value_counts()
```
6) Variabel X dan Y
```
X = df.drop(['Kategori'], axis=1)
y = df['Kategori']
```
7) Normalisasi data
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X = pd.DataFrame(scaler.fit_transform(X),columns=X.columns)
X
```
8) Membagi data *training* dan data *testing*
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
X_train.shape, X_test.shape
```
9) Menampilkan data *training*
```
X_train
```
10) Menampilkan data *testing*
```
X_test
```
11) Mengaktifkan model KNN

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=32,weights='distance',metric='euclidean')
knn.fit(X_train, y_train)
```

12) Menampilkan Prediksi yang dihasilkan dari Model KNN untuk Data *Testing*

```
y_pred = knn.predict(X_test)
y_pred
```

13) Menghitung akurasi model pada data *testing*

```
from sklearn.metrics import accuracy_score
print('Model accuracy score: {0:0.2f}'.format(accuracy_score(y_test, y_pred)))
```

14) Menghitung akurasi dari hasil prediksi pada data *training*

```
y_pred_train = knn.predict(X_train)
print('Training-set accuracy score: {0:0.2f}'.format(accuracy_score(y_train, y_pred_train)))
```

15) Menampilkan akurasi dari data *training* dan data *testing*

```
print('Training set score: {:.2f}'.format(knn.score(X_train, y_train)))
print('Test set score: {:.2f}'.format(knn.score(X_test, y_test)))
```

16) Menghitung kemunculan kategori pada data *testing*

```
y_test.value_counts()
```

17) Mengaktifkan *confusion matrix*

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm = confusion_matrix(y_test, y_pred)
```

18) Menampilkan *confusion matrix*

```
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=y_test.unique())
disp.plot(cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.show()
```

19) Menampilkan laporan evaluasi kalsifikasi

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

20) Menampilkan *cross-validation score* dari model

```
from sklearn.model_selection import cross_val_score
```

```
scores = cross_val_score(knn, X, y, cv = 10,
scoring='accuracy')
print('Cross-validation
scores:{}'.format(scores))
```

21) Menampilkan rata-rata *cross-validation score*

```
print('Average cross-validation score:
{:.2f}'.format(scores.mean()))
```

E. Jadwal Penelitian

Jadwal penelitian disajikan dalam bentuk tabel, sebagai berikut:

Tabel 4. Jadwal Penelitian

No	Nama Kegiatan	Tahun ke-								
		2023			2025			2026		
		Bulan ke-								
		3-6	7-9	10-12	10	11	12	1	2	3
1	**Tahap Persiapan**									
	a. Observasi									
	b. Rumusan masalah									
	c. Metode Penelitian									
	d. Uji Coba di Python									
	e. Seminar Proposal									
2	**Tahap Penelitian**									
	a. Pengambilan Data									
	b. Analisis Data									
	c. Penulisan Skripsi									
3	**Tahap Akhir**									
	a. Sidang Skripsi									
	b. Publikasi ilmiah									