



PENGGUNAAN ALGORITMA *K-NEAREST NEIGHBOR* (K-NN) UNTUK PREDIKSI CUACA DENGAN DATA RECORD

Rezano Verian Dwi Saputra¹, Ida Widaningrum², Jamilah Karaman³, Siti Lathifah Tsaqila³

^{1,2,3} Teknik Informatika Universitas Muhammadiyah Ponorogo

Jl Budi Utomo No 10 Ponorogo, Indonesia

⁴ Teknik Informatika Universitas Ahmad Dahlan

Jl. Ringroad Selatan, Kragilan, Tamanan, Kec. Banguntapan, Kabupaten Bantul, Daerah Istimewa
Yogyakarta 55191, Indonesia

Email : Rezanove19@gmail.com¹, iwidaningrum.as@gmail.com², jamilah@umpo.ac.id³,
sweety.lathifaatsaqila@gmail.com^{4*}

Dikimkan: 17 Desember 2023

Direvisi: 21 Agustus 2024

Diterima: 28 Oktober 2024

Abstrak

Desa Plancungan, Kecamatan Slahung, Kabupaten Ponorogo, merupakan penghasil tembakau, salah satu produk pertanian yang berperan penting dalam mendukung mata pencaharian, pertumbuhan ekonomi, dan penyerapan tenaga kerja. Di desa ini, petani membudidayakan tembakau rajangan, termasuk tembakau Virginia. Namun, proses penjemuran tembakau yang dilakukan di ruang terbuka selama dua hingga tiga hari sangat bergantung pada kondisi cuaca yang tidak menentu, yang sering kali memengaruhi kualitas hasil panen. Untuk mengatasi tantangan tersebut, penelitian ini menerapkan algoritma *K-Nearest Neighbor* (K-NN) guna memprediksi cuaca di Desa Plancungan. Data cuaca dikumpulkan menggunakan alat mikrokontroler yang dipasang di lokasi, yang merekam tiga parameter utama: suhu, kelembapan, dan tekanan udara. Dari total 10.000 data yang diperoleh, sebanyak 2.500 data digunakan untuk pelatihan dan pengujian algoritma. Hasil penelitian berupa prediksi cuaca—hujan, cerah, atau berawan—ditampilkan dalam bentuk halaman web. Informasi ini memungkinkan petani mendapatkan gambaran kondisi cuaca lebih awal, sehingga mereka dapat merencanakan langkah antisipasi untuk menjaga kualitas hasil panen dan mengoptimalkan proses penjemuran tembakau di tengah tantangan cuaca yang berubah-ubah.

Kata kunci: Prediksi Cuaca, K-Nearest Neighbor K-NN, Machine Learning, Mikrokontroler.

Abstract

Plancungan Village, Slahung District, Ponorogo Regency, is a tobacco producer, one of the agricultural products that plays an important role in supporting livelihoods, economic growth, and employment. In this village, farmers cultivate shredded tobacco, including Virginia tobacco. However, drying tobacco in the open for two to three days is highly dependent on uncertain weather conditions, which often affect the quality of the harvest. To overcome this challenge, this study applies the K-Nearest Neighbor (K-NN) algorithm to predict the weather in Plancungan Village. Weather data was collected using a microcontroller installed on site, which recorded three main parameters: temperature, humidity, and air pressure. Of the 10,000 data obtained, 2,500 were used for training and testing the algorithm. The results of the study in the form of weather predictions—rainy, sunny, or cloudy—were displayed in the form of a web page. This information allows farmers to get an overview of weather conditions earlier, so they can plan anticipatory steps to maintain the quality of their harvest and optimize the tobacco drying process amidst the challenges of changing weather..

Keyword: Weather Prediction, K-Nearest Neighbor K-NN, Machine Learning, Microcontroller.

PENDAHULUAN

Indonesia, sebagai negara agraris, memiliki sektor pertanian yang berperan penting tidak hanya sebagai sumber penghidupan, tetapi juga sebagai penopang pembangunan dan penyerap tenaga kerja dalam jumlah besar. Salah satu produk unggulan pertanian adalah tembakau, yang produksinya tersebar di berbagai wilayah, termasuk Desa Plancungan, Kecamatan Slahung, Kabupaten Ponorogo. Petani di desa ini membudidayakan tembakau rajangan dengan varietas seperti tembakau Virginia, yang pertumbuhannya sangat dipengaruhi oleh faktor iklim dan letak geografis. Namun, perubahan cuaca yang tidak menentu sering menjadi kendala bagi petani, terutama dalam proses penjemuran tembakau yang dilakukan di ruang terbuka. Penjemuran yang ideal membutuhkan waktu 2-3 hari di bawah sinar matahari, tetapi jika tembakau terkena air, kualitasnya dapat menurun dengan perubahan warna menjadi lebih gelap. Sayangnya, petani tembakau di Desa Plancungan masih mengandalkan perkiraan manual untuk menentukan cuaca, sementara akses informasi cuaca yang akurat sering kali terbatas.

Untuk membantu petani tembakau di Desa Plancungan, dikembangkan sebuah aplikasi peramalan cuaca berbasis web menggunakan algoritma machine learning *K-Nearest Neighbor* (K-NN). Data cuaca diperoleh melalui mikrokontroler yang dipasang di lokasi tertentu, mencatat parameter seperti suhu (temperature), kelembapan udara (humidity), dan tekanan udara (pressure). Berdasarkan data historis tersebut, kondisi cuaca di masa mendatang dapat diprediksi secara akurat. Informasi ini sangat berguna bagi petani untuk mengetahui kondisi cuaca, baik saat ini maupun beberapa waktu ke depan, sehingga mereka dapat merencanakan penjemuran tembakau dengan lebih efektif. Dengan demikian, aplikasi ini diharapkan dapat meminimalkan dampak negatif dari cuaca yang tidak menentu, serta mendukung produktivitas petani tembakau di wilayah tersebut.

Machine learning (ML) merupakan cabang kecerdasan buatan yang memungkinkan komputer belajar dari data tanpa perlu diprogram ulang secara manual. Dengan memanfaatkan algoritma dan data, ML mampu membuat prediksi atau klasifikasi berdasarkan pola yang ditemukan dalam data historis. Proses kerja ML melibatkan pengumpulan data, eksplorasi fitur, pemilihan model, pelatihan, hingga evaluasi hasil. Komponen penting seperti algoritma, kumpulan data, representasi fitur, dan target pembelajaran menjadi elemen kunci dalam keberhasilan implementasi ML. Metode yang digunakan dapat disesuaikan dengan jenis data dan tujuan, seperti *supervised learning* untuk memprediksi hasil spesifik dengan data berlabel, atau *unsupervised learning* untuk menemukan pola tersembunyi dalam data. Keunggulan ML adalah kemampuannya beradaptasi dengan berbagai jenis data dan aplikasinya yang luas, termasuk dalam bidang prediksi cuaca.

Dalam prediksi cuaca, machine learning digunakan untuk menganalisis data historis seperti suhu, kelembapan, tekanan udara, dan kecepatan angin guna memprediksi kondisi cuaca di masa depan. Salah satu algoritma yang sering digunakan adalah *K-Nearest Neighbor* (K-NN), yang bekerja dengan membandingkan data cuaca baru dengan data historis yang memiliki pola serupa. Algoritma ini termasuk dalam kategori *supervised learning*, yang membutuhkan data berlabel seperti kondisi cerah, hujan, atau berawan untuk melatih model prediksi. Proses ini melibatkan sensor atau perangkat mikrokontroler untuk mengumpulkan data cuaca secara real-time, yang kemudian diolah dan digunakan dalam model K-NN. Hasil prediksi disajikan melalui antarmuka digital, seperti halaman web atau aplikasi, sehingga pengguna dapat mengakses informasi cuaca yang akurat. Dengan prediksi ini, petani atau pihak lain dapat merencanakan kegiatan yang bergantung pada cuaca, seperti penjemuran atau panen, sehingga mengurangi dampak negatif dari cuaca yang tidak menentu.

Penelitian sebelumnya telah banyak membahas prediksi cuaca menggunakan berbagai metode. Salah satunya adalah metode *fuzzy c-means* (FCM) yang dikombinasikan dengan *fuzzy Sugeno* untuk memprediksi curah hujan berdasarkan data cuaca dari BMKG Bandung [1]. Hasil penelitian ini mencapai akurasi 71,10% dengan memanfaatkan suhu, kelembapan, dan kecepatan angin sebagai input. Ada pula penelitian yang menggunakan algoritma *K-Nearest Neighbor* (K-NN) untuk memprediksi perubahan suhu tahunan di Indonesia dengan data historis selama 10 tahun, yang memperhitungkan variabel seperti curah hujan dan rata-rata suhu tahunan [2]. Penelitian lainnya memanfaatkan K-NN untuk klasifikasi potensi hujan harian, di mana data dikumpulkan menggunakan sensor mikrokontroler [3]. Sebagai perbandingan, model regresi linier juga digunakan untuk prakiraan cuaca, meski akurasi yang dihasilkan hanya mencapai 70%, menunjukkan bahwa pendekatan berbasis K-NN memiliki potensi lebih tinggi dalam beberapa kasus [4]. [5] memanfaatkan algoritma untuk memprediksi tingkat curah hujan dan suhu pada tahun tertentu, sehingga petani dapat memperoleh informasi lebih awal dan mengambil langkah mitigasi jika diperlukan. Hasil penelitian menunjukkan bahwa algoritma KNN mampu menyediakan data terkait curah hujan dan suhu yang relevan untuk kebutuhan tanaman padi.

Penelitian yang dilakukan mengimplementasikan *machine learning* berbasis algoritma K-NN untuk memprediksi cuaca di Desa Plancungan, Kecamatan Slahung. Data dikumpulkan melalui sensor DHT11 dan BMP280 yang merekam suhu, kelembapan, dan tekanan udara. Dengan data ini, prediksi cuaca seperti cerah, hujan, atau berawan akan disajikan dalam bentuk antarmuka berbasis web, memberikan informasi hingga 1–2 jam ke depan. Pendekatan ini dirancang untuk membantu petani tembakau yang sangat bergantung pada sinar matahari dalam proses penjemuran. Dengan informasi cuaca yang akurat, petani dapat merencanakan aktivitas mereka dengan lebih baik, mengurangi risiko kerusakan tembakau akibat perubahan cuaca yang tidak terduga.

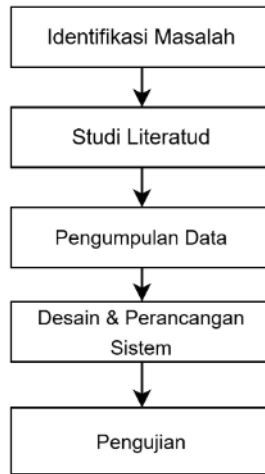
Penelitian sebelumnya telah banyak mengembangkan sistem otomatisasi pada atap penjemur dengan memanfaatkan berbagai sensor. Salah satu contohnya adalah sistem pembuka atap otomatis berbasis mikrokontroler Arduino Uno ATmega328, yang dilengkapi dengan sensor cahaya LDR, sensor hujan LY-83, dan tiga toggle switch untuk kontrol manual [6]. Sistem ini dirancang untuk membantu peternak sapi dalam menjemur sapi mereka tanpa perlu mengeluarkannya satu per satu. Sistem ini bekerja secara otomatis berdasarkan kondisi cuaca: atap akan terbuka saat cerah dan tertutup saat mendung atau hujan. Selain itu, mode manual juga memungkinkan pengguna mengontrol atap secara langsung, terlepas dari input sensor.

Penelitian lainnya memanfaatkan teknologi serupa untuk kebutuhan lain, seperti pengeringan gabah [7] dan jagung [8]. Pada penjemur gabah [9], sistem menggunakan sensor hujan YL83 dan sensor cahaya untuk menggerakkan motor servo, sementara pada penjemur jagung, kontrol atap dilakukan menggunakan modul raindrop, mikrokontroler Wemos D1 Mini R2 berbasis IoT, serta aplikasi smartphone. Penelitian pada penjemur kerupuk bahkan menambahkan fitur seperti LCD dan module Bluetooth untuk memberikan informasi lebih rinci kepada pengguna. Dalam konteks penjemuran tembakau, penelitian ini menggunakan NodeMCU ESP8266 dengan sensor cahaya LDR, sensor hujan MD-0127, dan sensor tekanan udara BMP280 untuk menggerakkan Twin Gear Box Motor DC. Sistem ini dirancang untuk mengontrol atap penjemur tembakau yang akan bergerak membuka saat mendeteksi cahaya matahari dan menutup saat hujan atau menjelang malam hari. Semua elemen sistem ditempatkan dalam miniatur lahan dengan struktur menyerupai atap rumah, memberikan solusi efisien dan otomatis bagi petani tembakau.

METODE PENELITIAN

1. Tahapan Penelitian

Tahapan penelitian melalui beberapa tahapan seperti pada gambar 1.



Gambar 1. Tahapan Penelitian

Pada tahap **Identifikasi Masalah**, penelitian ini mengatasi kurangnya informasi cuaca akurat bagi petani tembakau di Desa Plancangan. Petani menghadapi risiko hujan mendadak saat menjemur tembakau di area terbuka. Peneliti merancang sistem berbasis *machine learning* menggunakan algoritma KNN berbasis web, yang memprediksi cuaca dengan variabel suhu, kelembapan, dan tekanan udara. Data dikumpulkan melalui mikrokontroler, diproses, dan disimpan dalam basis data, dengan output berupa hujan, cerah, atau berawan.

Tahap **Studi Literatur** melibatkan kajian sumber-sumber seperti jurnal, artikel, buku, dan situs internet terkait metode dan objek penelitian. Kajian ini memberikan dasar teori dan mempermudah pelaksanaan penelitian.

Pada tahap **Pengumpulan Data**, data dikumpulkan dari referensi yang relevan serta pengukuran cuaca di Desa Plancangan menggunakan sensor DHT11 dan BMP280. Sensor ini mencatat suhu, kelembapan, dan tekanan udara sebagai input untuk sistem prediksi

2. Desain dan Perancangan Sistem



Gambar 2. Flowchart Sistem

Flowchart sistem digunakan untuk menggambarkan secara rinci proses dan hubungan antarproses dalam sebuah program melalui diagram yang saling terhubung. Penjelasan terkait alur sistem pada Gambar 2 adalah sebagai berikut:

Perekaman Data: Data cuaca dicatat menggunakan sensor DHT11 dan BMP280. Sensor DHT11 mengukur suhu (*temperature*) dan kelembapan (*humidity*), sedangkan BMP280 mendeteksi tekanan udara (*pressure*) sekaligus menghitung ketinggian objek. Data yang direkam menggambarkan kondisi cuaca di Desa Plancungan.

Proses Pelatihan K-NN: Algoritma K-Nearest Neighbor (K-NN) dilatih menggunakan data yang tersedia. Algoritma ini mengklasifikasikan data baru berdasarkan kedekatan atau kesamaan dengan data sebelumnya, mengelompokkan data baru ke dalam kategori tertentu.

Prediksi Cuaca: Hasil pelatihan K-NN menghasilkan prediksi kondisi cuaca seperti hujan, berawan, atau cerah. Informasi ini membantu petani dalam mengambil keputusan terkait aktivitas harian mereka.

K-Nearest Neighbor (KNN) adalah algoritma supervised learning yang mengklasifikasikan query instance berdasarkan mayoritas kategori dalam tetangga terdekatnya. Tujuan algoritma ini adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan data pelatihan yang telah tersedia. Algoritma ini tidak membangun model tertentu, melainkan hanya mengandalkan memori data. Ketika diberikan titik query, algoritma mencari sejumlah k titik pelatihan terdekat dengan query tersebut. Klasifikasi dilakukan dengan cara memilih kategori mayoritas dari k titik tersebut melalui proses voting. Algoritma K-NN menentukan nilai prediksi dari query instance baru berdasarkan klasifikasi dari tetangga terdekatnya [10], [11].

Metode K-NN bekerja dengan cara sederhana, menggunakan jarak terpendek antara query instance dan data pelatihan untuk menentukan tetangga terdekat (*nearest neighbors*). Data pelatihan dipetakan ke dalam ruang berdimensi banyak, di mana setiap dimensi merepresentasikan fitur data. Ruang ini terbagi menjadi bagian-bagian berdasarkan klasifikasi data pelatihan. Sebuah titik dalam ruang tersebut akan diberi kelas c jika kelas c merupakan kategori yang paling sering muncul di antara k tetangga terdekat titik tersebut. Jarak antara titik-titik biasanya dihitung menggunakan *Euclidean Distance*, yang dirumuskan sebagai berikut:

$$D(a, b) = \sqrt{\sum_{k=1}^d (a_k - b_k)^2}, \quad (1)$$

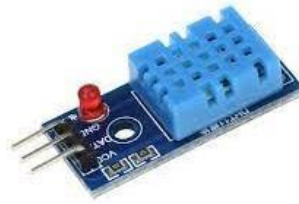
Di mana $D(a,b)$ adalah jarak skalar antara dua vektor a dan b , dengan ukuran dimensi d .

Selama fase pelatihan, algoritma hanya menyimpan vektor fitur dan klasifikasi data pelatihan. Kemudian, pada fase klasifikasi, fitur yang sama dihitung untuk data uji (yang klasifikasinya belum diketahui). Jarak antara vektor baru dan seluruh vektor pelatihan dihitung, lalu sejumlah k tetangga terdekat dipilih. Klasifikasi data baru ditentukan berdasarkan kategori yang paling banyak muncul di antara tetangga-tetangga tersebut.

3. Sensor DHT11

DHT11 adalah sensor digital untuk suhu dan kelembapan dengan keandalan tinggi, stabilitas jangka panjang, dan respons cepat (lihat Gambar 3). Sensor ini dilengkapi dengan mikrokontroler 8-bit, mendukung komunikasi melalui port serial satu kabel, dan memiliki jangkauan transmisi hingga 20 meter. Akurasi pengukurannya mencapai $\pm 2^\circ\text{C}$ untuk suhu dan $\pm 5\%$ untuk kelembapan. Dengan desain yang ringkas, konsumsi daya rendah, dan konektor 4 pin yang mudah digunakan, DHT11

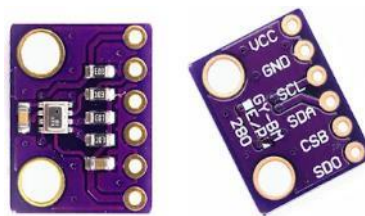
sangat ideal untuk berbagai aplikasi yang memerlukan pengukuran suhu dan kelembapan secara efektif [12], [13].



Gambar 3. Sensor DHT11

4. Sensor Tekanan Udara BMP280

Sensor tekanan udara BMP280 (gambar 4) adalah sensor barometrik berukuran kecil dengan konsumsi daya rendah, ideal untuk perangkat portabel seperti ponsel, modul GPS, dan jam tangan. Menggunakan teknologi piezo-resistif dari Bosch, sensor ini menawarkan akurasi tinggi, linearitas, dan stabilitas. Dengan akurasi tekanan absolut sekitar ± 1 Pa dan suhu $\pm 1^\circ\text{C}$, sensor ini juga dapat berfungsi sebagai altimeter dengan tingkat akurasi ± 1 meter, berkat kemampuannya mengukur tekanan yang bervariasi sesuai ketinggian. [14], [15].



Gambar 4. Sensor BMP280

5. Rancangan Pengujian

Pengujian sistem machine learning ini bertujuan untuk mengukur akurasi algoritma K-Nearest Neighbor (K-NN) dalam memprediksi cuaca di Desa Plancungan. Pengujian dilakukan dengan menggunakan dataset yang diambil dari 10.000 data dan dilatih pada 2.500 data yang mencakup suhu, kelembapan, dan tekanan udara. Data dikumpulkan menggunakan sensor DHT11 dan BMP280, lalu dikirim ke server untuk disimpan dalam database.

Proses machine learning melibatkan perhitungan nilai k dan jarak antara data training dan dataset, dengan input berupa suhu, kelembapan, dan tekanan udara. Output sistem memprediksi kondisi cuaca—cerah, berawan, atau hujan—di wilayah tersebut, yang kemudian ditampilkan pada sebuah web. Pengujian dilakukan dalam dua skenario, yaitu prediksi untuk satu jam ke depan dan dua jam ke depan. Hasil dari kedua prediksi dibandingkan untuk mengevaluasi konsistensi dan keakuratan sistem.

HASIL DAN PEMBAHASAN

1. Persiapan Alat

Langkah pertama adalah mempersiapkan perangkat untuk merekam data. Mikrokontroler dirancang dan dirakit sebagaimana ditunjukkan pada Gambar 5, untuk mengumpulkan data dari wilayah Desa Plancungan, mencakup suhu, kelembapan, dan tekanan udara. NodeMCU ESP8266

digunakan sebagai pengendali utama, bekerja sama dengan sensor DHT11 dan BMP280 yang dirangkai secara terintegrasi untuk memastikan akurasi pengukuran di lokasi tersebut.



Gambar 5. Mikrokontroler yang sudah dirangkai

Selanjutnya, kode program disusun (Gambar 6) dan diunggah ke mikrokontroler. Program ini berfungsi mengontrol sensor DHT11 dan BMP280 agar dapat mengukur parameter lingkungan secara akurat pada waktu tertentu. Data hasil pengukuran kemudian disimpan dalam database untuk keperluan analisis lebih lanjut.

```

firmware | Arduino 1.8.19
File Edit Sketch Tools Help

firmware

unsigned status;
status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
}

void loop() {
  int rain = analogRead(RAIN);
  bool light = digitalRead(LIGHT);
  sensors_event_t temp_event, pressure_event;
  bmp_temp->getEvent(&temp_event);
  bmp_pressure->getEvent(&pressure_event);

  float bmp_temp = temp_event.temperature;
  float bmp_pressure = pressure_event.pressure;
  delay(1000);

  sensors_event_t event;
  dht.temperature().getEvent(&event);

  float dht_temp = event.temperature;
  float dht_hum = event.relative_humidity;
  }

```

Gambar 6. Coding yang di gunakan untuk mikrokontroler

2. Record Data dan Import Data

```

COM7
-----
RAIN      : 8
LIGHT     : 0
TEMPBMP   : 30.24 *C
PRES      : 997.80 hPa
TEMPDHT   : 28.90 *C
HUM       : 28.90%
[HTTP] begin...
[HTTP] POST...
[HTTP] POST... code: 200
received payload:
<<
{"status": "success"}
>>
RAIN      : 13
LIGHT     : 0
TEMPBMP   : 30.24 *C
PRES      : 997.72 hPa
TEMPDHT   : nan *C
HUM       : nan%
[HTTP] begin...
[HTTP] POST...
[HTTP] POST... code: 500
RAIN      : 10
LIGHT     : 0
TEMPBMP   : 30.25 *C
PRES      : 997.71 hPa
TEMPDHT   : 28.90 *C
HUM       : 28.90%
 Autoscrol  Show timestamp

```

Gambar 7. Proses data yang di ambil dari sensor

Data yang dikumpulkan meliputi suhu (temperature), kelembapan udara (humidity), dan tekanan udara (pressure) menggunakan mikrokontroler Arduino. Data ini memiliki sejumlah atribut, yaitu id, timestamp, temperature, humidity, pressure, dan condition, seperti terlihat pada Gambar 7. Pengumpulan data dilakukan secara berkala setiap 3 detik.

Setelah direkam, data dikirim ke server dan disimpan dalam database dengan struktur yang telah terorganisir sesuai label kolom, sebagaimana diperlihatkan pada Gambar 8.

	id	timestamp	temp	hum	light	rain	pressure	cond
Ubah Salin Hapus	77115	2023-01-04 20:43:20	31.32	29.3	1	13	997.93	CLEAR
Ubah Salin Hapus	77114	2023-01-04 20:43:18	31.32	29.3	1	15	997.94	CLEAR
Ubah Salin Hapus	77113	2023-01-04 20:43:15	31.32	29.3	1	12	997.94	CLEAR
Ubah Salin Hapus	77112	2023-01-04 20:43:13	31.32	29.3	1	10	997.95	CLEAR
Ubah Salin Hapus	77111	2023-01-04 20:43:10	31.33	29.3	1	13	997.95	CLEAR
Ubah Salin Hapus	77110	2023-01-04 20:43:07	31.32	29.3	1	6	997.94	CLEAR
Ubah Salin Hapus	77109	2023-01-04 20:43:05	31.32	0	1	16	997.95	CLEAR
Ubah Salin Hapus	77108	2023-01-04 20:43:02	31.33	29.3	1	15	997.95	CLEAR
Ubah Salin Hapus	77107	2023-01-04 20:43:00	31.32	0	1	7	997.94	CLEAR
Ubah Salin Hapus	77106	2023-01-04 20:42:57	31.32	29.3	1	9	997.96	CLEAR
Ubah Salin Hapus	77105	2023-01-04 20:42:54	31.32	29.3	1	4	997.97	CLEAR
Ubah Salin Hapus	77104	2023-01-04 20:42:52	31.33	29.3	1	3	997.95	CLEAR
Ubah Salin Hapus	77103	2023-01-04 20:42:49	31.32	28.3	1	1	997.95	CLEAR
Ubah Salin Hapus	77102	2023-01-04 20:42:47	31.32	0	1	9	997.96	CLEAR
Ubah Salin Hapus	77101	2023-01-04 20:42:44	31.32	0	1	15	997.93	CLEAR
Ubah Salin Hapus	77100	2023-01-04 20:42:41	31.32	0	1	29	997.93	CLEAR
Ubah Salin Hapus	77099	2023-01-04 20:42:39	31.32	0	1	18	997.95	CLEAR
Ubah Salin Hapus	77098	2023-01-04 20:42:36	31.33	29.3	1	9	997.96	CLEAR
Ubah Salin Hapus	77097	2023-01-04 20:42:34	31.33	0	1	18	997.96	CLEAR

Gambar 8. Hasil record masuk ke database

3. Import Library

Python dimulai dengan menggunakan pustaka pandas, numpy, dan scikit-learn untuk mengolah data dan membangun model pembelajaran mesin seperti diperlihatkan pada gambar 9. Kode ini bertujuan untuk mempersiapkan data mentah agar siap digunakan dalam analisis atau pembelajaran mesin. Proses dimulai dengan mengimpor beberapa modul, termasuk MinMaxScaler untuk normalisasi data, KNeighborsClassifier untuk klasifikasi berbasis algoritma K-Nearest Neighbors, serta train_test_split untuk membagi dataset menjadi data latih dan uji. Modul lokal bernama records digunakan untuk mengambil data dalam format dictionary dengan batasan maksimum 10.000 entri, yang kemudian dikonversi menjadi DataFrame menggunakan fungsi pd.DataFrame.from_dict. Setelah itu, kolom id

dihapus dari DataFrame, dan data waktu yang terdapat dalam kolom timestamp dipecah menjadi elemen bulan (month), hari (day), dan jam (hour) dengan memanfaatkan fungsi apply dari pandas. Setelah pemrosesan selesai, kolom timestamp dihapus untuk menjaga data tetap bersih. Selain itu, kode ini juga mengabaikan peringatan dengan menggunakan warnings.filterwarnings('ignore') agar tidak mengganggu jalannya program..

```
from flask import *
from model import records
from predictor import api_predict
import datetime

from sklearn.preprocessing import MinMaxScaler
import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from model import records
import warnings
warnings.filterwarnings('ignore')
```

Gambar 9. Import library

4. Pra-proses dan Proses Data (*Processing Data*)

Proses yang ditampilkan dalam gambar 10 menunjukkan pelatihan dan evaluasi model K-Nearest Neighbors (KNN) pada dataset tertentu, termasuk penghitungan akurasi dan eksplorasi nilai parameter `n_neighbors`. Dataset dibagi menjadi data latih dan data uji menggunakan fungsi `train_test_split` dengan proporsi 75% untuk data latih dan 25% untuk data uji, serta nilai `random_state=25` untuk memastikan hasil yang konsisten. Model KNN diinisialisasi dengan parameter `n_neighbors=4`, yang berarti model akan mempertimbangkan empat tetangga terdekat dalam menentukan kelas prediksi. Setelah itu, model dilatih menggunakan data latih dan prediksi dilakukan pada data uji, menghasilkan label prediksi yang disimpan dalam variabel `y_pred`. Evaluasi model dilakukan dengan menggunakan tiga metrik utama: laporan klasifikasi yang mencakup *precision*, *recall*, dan *f1-score*; matriks kebingungan untuk menganalisis kesalahan prediksi; serta akurasi model yang dihitung dengan metode *score*.

Selanjutnya, dilakukan eksperimen untuk mengevaluasi performa model KNN dengan berbagai nilai parameter `n_neighbors` dari 1 hingga 39. Dalam setiap iterasi, model KNN dilatih dengan data latih dan dievaluasi menggunakan data uji. Error rate dihitung sebagai rata-rata kesalahan prediksi terhadap label sebenarnya, dan hasilnya disimpan dalam list `error_rate`. Eksperimen ini memberikan wawasan tentang pengaruh nilai `n_neighbors` terhadap performa model. Di bagian bawah kode, terdapat fungsi `api_predict` untuk memproses data input dalam format API agar sesuai dengan format yang diperlukan oleh model prediksi. Secara keseluruhan, kode ini mencakup proses pembagian dataset, pelatihan model KNN, evaluasi performa model, dan eksperimen parameter untuk menemukan konfigurasi terbaik, memastikan bahwa model bekerja secara optimal dengan data yang diberikan.

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=25)

knn = KNeighborsClassifier(n_neighbors= 4)
knn.fit(X_train, y_train)

y_pred = knn.predict(X_test)
```

Gambar 10. Proses KNN

Selain itu, evaluasi algoritma dapat menggunakan *confusion matrix*, *precision*, *recall* dan *f1 score*. Metode *confusion_matrix* dan *classification_report* pada *sklearn.metrics* dapat digunakan untuk menghitung matriks. Untuk tahapan mengevaluasi algoritma bisa dilihat pada Gambar 11.

```
#evaluating the algorithm
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
```

Gambar 11. Proses evaluasi algoritma

5. Tahap Pengujian dan Hasil Klasifikasi

Tahap terakhir yaitu tahapan pengujian dan hasil klasifikasi. Proses klasifikasi menggunakan jangka waktu 1 jam dan 2 jam kedepan dilihat pada Gambar 12 dan Gambar 13 yang mendapatkan hasil yang sama.

```
return [dict(row) for row in result]

def get_date(date):
    switcher = 1
    query = sa.select([records.table]).where(sa.cast(records.table.c.timestamp, sa.Date) == date).order_by(records.table.c.timestamp.desc)
    result = conn.execute(query).fetchall()

    if result is None:
        return None
    else:
        #use row hour as key
        data = (row['timestamp']).hour: dict(row) for row in result
        if switcher == 1:
            #condense data to 1 hour interval
            condensed_data = {}
            for hour in data:
                if hour % 1 == 0:
                    #average the data
                    condensed_data[hour] = data[hour]
                    for key in data[hour]:
                        if key not in ['id', 'timestamp', 'cond']:
                            try:
                                condensed_data[hour][key] = (data[hour][key] + data[hour + 1][key]) / 1
                            except:
```

Gambar 12. Membuat prediksi dalam jangka waktu 1 jam kedepan

	precision	recall	f1-score	support
CLEAR	0.95	1.00	0.97	1161
CLOUDY	1.00	0.95	0.97	1339
accuracy			0.97	2500
macro avg	0.97	0.97	0.97	2500
weighted avg	0.97	0.97	0.97	2500

Gambar 13. Hasil klasifikasi prediksi jangka 1 jam sama dengan jangka waktu 2 jam

Hasil tersebut menunjukkan bahwa algoritma K-NN mampu mengklasifikasikan 2.500 data dari total 10.000 dataset dalam set uji yang digunakan untuk menentukan kondisi cuaca, dengan tingkat akurasi mencapai 97%. Prediksi yang dihasilkan menunjukkan performa yang baik, terutama dalam mengidentifikasi kondisi berawan (cloudy), yang sesuai dengan keadaan di wilayah Desa Plancungan.

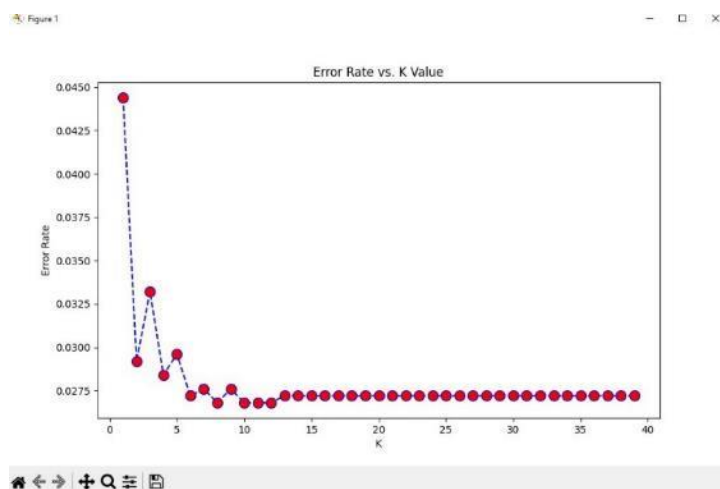
6. Comparing Error Rate with the K value

Kemudian dilakukan perbandingan untuk mencari tingkat akurasi terbaik saat nilai K (neighbor terdekat) secara acak, di bagian pelatihan dan prediksi diatur ke k=4 dan menghasilkan akurasi 97% Selain itu, salah satu cara untuk menemukan nilai K terbaik adalah dengan memplot sebidang nilai K dan tingkat kesalahan yang sesuai untuk kumpulan data. Pada bagian ini, plot kesalahan standar dari nilai prediksi kumpulan pengujian untuk semua nilai k antara 1 dan 40. Proses untuk mencari tingkat akurasi bisidilihat pada Gambar 14.

```
for i in range(1,40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    pred_i = knn.predict(X_test)
    error_rate.append(np.mean(pred_i != y_test))
```

Gambar 14. Mencari tingkat akurasi nilai k

Langkah selanjutnya adalah memplot nilai error terhadap nilai k. Akan muncul grafik seperti gambar 15.



Gambar 15. Grafik perbandingan tingkat akurasi k 1-40

Pada Gambar 15. grafik menunjukkan bahwa nilai k-1 dapat memprediksi dengan benar semua data uji, meskipun nilai k selalu lebih kecil dari nilai tersebut. Untuk tetangga terdekat nilai k yaitu berada pada nilai k-7 dan k-8. Kemudian untuk dapat dilihat bahwa yang memiliki tingkat kesalahan rata-rata (*mean error*) sama dengan 0 berada pada K-10 sampai k-12.

7. Perbandingan Prediksi *Machine Learning* Dengan Aplikasi Cuaca

Pada tahap ini mencoba membandingkan apakah hasil prediksi antara *Machine Learning* dengan aplikasi cuaca pada handphone memiliki kesamaan atau tidak. Perbandingan dilakukan pada tanggal 9 Februari 2023. Hasil dari kedua perbandingan tersebut dapat dilihat pada Gambar 26. Sampai 29.

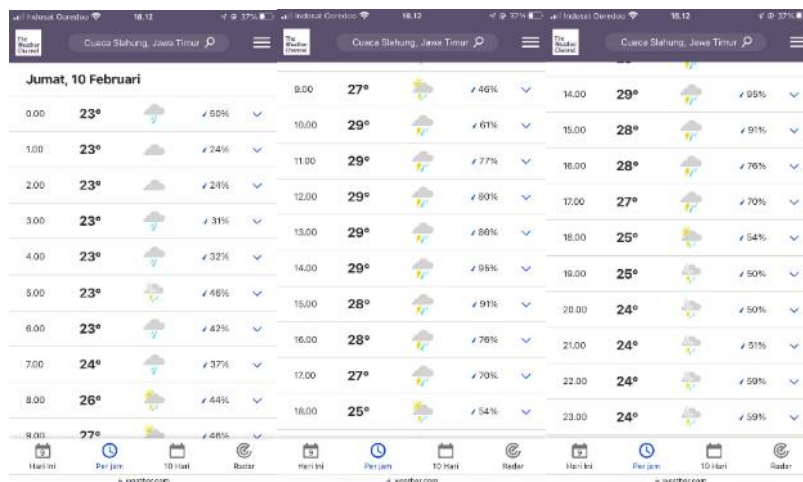
2023-02-10					2023-02-09 Prediksi				
Jam	Kelembaban	Tekanan Atm	Suhu	Kondisi	Jam	Kelembaban	Tekanan Atm	Suhu	Kondisi
00	22.42%	990.5 hPa	22.42°C	Cerah	00	23.39%	996.49 hPa	23.39°C	Berawan
01	23.75%	991.33 hPa	23.75°C	Cerah	01	23.05%	1011.87 hPa	23.05°C	HujanY
02	23.34%	1010.91 hPa	23.34°C	Berawan	02	22.32%	990.9 hPa	22.32°C	Berawan
03	25.04%	1011.36 hPa	25.04°C	Cerah	03	24.02%	991.37 hPa	24.02°C	Cerah
04	25.07%	1016.95 hPa	25.07°C	HujanY	04	26.27%	1011.72 hPa	26.27°C	Berawan
05	25.57%	1007.42 hPa	25.57°C	Cerah	05	24.17%	1005.28 hPa	24.17°C	Cerah
06	25.75%	997.46 hPa	25.75°C	Cerah	06	26.13%	1006.81 hPa	26.13°C	HujanY
07	24.22%	1011.06 hPa	24.22°C	Berawan	07	28.02%	1003.79 hPa	28.02°C	HujanY
08	26.69%	1017.31 hPa	26.69°C	Berawan	08	27.69%	1017.45 hPa	27.69°C	HujanY
09	27.54%	1005.32 hPa	27.54°C	Cerah	09	30.11%	1018.49 hPa	30.11°C	Cerah
10	25.08%	1009.26 hPa	25.08°C	Cerah	10	27.92%	1004.47 hPa	27.92°C	Cerah
11	28.15%	1000.93 hPa	28.15°C	Berawan	11	31.19%	100.73 hPa	31.19°C	HujanY
12	30.61%	1013.67 hPa	30.61°C	HujanY	12	30.2%	994.75 hPa	30.2°C	Cerah
13	28.17%	993.89 hPa	28.17°C	HujanY	13	31.74%	1009.25 hPa	31.74°C	Berawan
14	27.8%	1000.99 hPa	27.8°C	Berawan	14	28.89%	1001.06 hPa	28.89°C	HujanY
15	26.96%	1019.27 hPa	26.96°C	HujanY	15	26.65%	1000.86 hPa	26.65°C	HujanY
16	27.05%	1016.29 hPa	27.05°C	Berawan	16	26.92%	1015.74 hPa	26.92°C	Cerah
17	29.38%	1013.6 hPa	29.38°C	HujanY	17	25.99%	1016.46 hPa	25.99°C	Berawan
18	27.4%	991.2 hPa	27.4°C	Cerah	18	28.92%	1013.77 hPa	28.92°C	HujanY
19	27.49%	1001.46 hPa	27.49°C	Berawan	19	27.43%	990.38 hPa	27.43°C	Berawan
20	25.58%	1015.69 hPa	25.58°C	Berawan	20	27.19%	1014.77 hPa	27.19°C	Berawan
21	24.23%	999.04 hPa	24.23°C	Cerah	21	26.28%	996.44 hPa	26.28°C	Berawan
22	24.79%	1014 hPa	24.79°C	HujanY	22	26.15%	1013.26 hPa	26.15°C	HujanY
23	25.74%	996.82 hPa	25.74°C	HujanY	23	27.69%	1003.31 hPa	27.69°C	Berawan

Gambar 16. Hasil prediksi per jam dengan mesin learning

Tanggal	Kelembaban	Tekanan Atm	Suhu	Kondisi
Kamis, 9/2/2023	74.62%	1004.95 hPa	26.98°C	Berawan
Jumat, 10/2/2023	67.31%	1006.07 hPa	26.15°C	Hujan
Sabtu, 11/2/2023	82.74%	1003.36 hPa	27.35°C	Berawan
Minggu, 12/2/2023	89.24%	1006.41 hPa	26.51°C	Berawan
Senin, 13/2/2023	70.64%	1005.65 hPa	27.93°C	Hujan
Selasa, 14/2/2023	75.91%	1006.21 hPa	29.57°C	Hujan
Rabu, 15/2/2023	70.34%	1019.79 hPa	27.50°C	Hujan
Kamis, 16/2/2023	70.26%	1004.23 hPa	28.19°C	Berawan
Jumat, 17/2/2023	80.24%	1006.21 hPa	28.23°C	Cerah
Sabtu, 18/2/2023	70.53%	1005.32 hPa	29.72°C	Berawan
Minggu, 19/2/2023	80.24%	1004.36 hPa	27.84°C	Berawan

Gambar 17. Data harian prediksi cuaca





Gambar 18. Prediksi per jam pada aplikasi cuaca hp



Gambar 19. Hasil prediksi harian pada aplikasi cuaca di hp

Dari hasil perbandingan antara *Machine Learning* dengan aplikasi cuaca pada hp, memiliki perbedaan segi tampilan, suhu udara, tekanan udara, kelembaban udara, prediksi harian 10 hari, dan ada beberapa kesamaan prediksinya di hari yang sama yaitu pada tanggal 9 Februari 2023 sampai tanggal 10 Februari 2023, dan prediksi harian 10 hari. Pada mesin learning prediksi harian selama 10 hari, cuaca hujan terletak pada tanggal 10 Februari 2023, tanggal 13 Februari 2023, tanggal 14 Februari 2023, dan tanggal 15 Februari 2023. Untuk kondisi harian berawan terletak pada tanggal 9 Februari 2023, tanggal 11 Februari 2023, tanggal 12 Februari 2023, tanggal 16 Februari 2023, tanggal 18 Februari 2023, dan tanggal 19 Februari 2023. Untuk kondisi harian cuaca cerah terletak pada tanggal 17 Februari 2023 bisa dilihat pada Gambar 27. Sedangkan pada aplikasi cuaca pada handphone ramalan 10 hari kedepan menghasilkan kondisi cuaca hujan dari tanggal 9 Februari sampai tanggal 19 Februari, bisa dilihat pada Gambar 29. Untuk prediksinya per jam juga memiliki perbedaan bisa dilihat pada Gambar 26 dan Gambar 28. Pada aplikasi cuaca hp tanggal 9 Februari 2023 pukul 18.00-

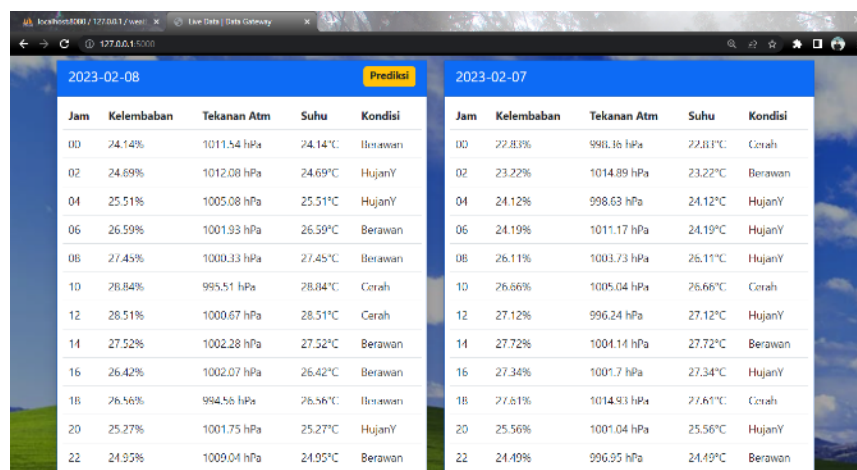
20.00 kondisi cuaca berawan, pada jam berikutnya pukul 21.00–23.00 kondisi cuaca hujan dilihat pada Gambar 28. Sedangkan pada mesin learning tanggal 9 Februari 2023 pukul 18.00 memiliki kesamaan prediksi yaitu hujan, selanjutnya pukul 19.00-21.00 dan pukul 23.00 kondisi cuaca berawan selanjutnya pukul 22.00 meprediksi hujan, dilihat pada Gambar 26. Untuk prediksi pada tanggal 10 Februari 2023 pada aplikasi cuaca hp kondisi hujan di perkirakan terjadi hampir sehari dari pukul 00.00-23.00 bisa dilihat pada Gambar 28. Pada mesin learning tanggal 10 Februari 2023 diperkirakan kondisi hujan pada pukul 04.00, pukul 12.00-13.00, pukul 15.00, pukul 17.00, pukul 22.00-23.00 dapat dilihat pada Gambar 26.

8. Tampilan Interface Web

Pada gambar di bawah ini merupakan tampilan interface hasil prediksi pada tanggal 7 Februari 2023 sampai tanggal 8 Februari 2023 menggunakan *machine learning* algoritma *K-Nearest Neighbor* K-NN yang di tampilkan pada halaman web browser, untuk pengalaman prediksi pertama yang dibuat dalam satu jam berikutnya dapat dilihat pada Gambar 4.22, kemudian lanjutkan memprediksi selama dua jam kedepan dapat dilihat pada Gambar 4.23. Dari kedua hasil perbandingan dapat kita lihat bahwa untuk tampilan prediksi jangka waktu dua jam lebih ringkas karena prediksi yang di tampilkan menggunakan jangka waktu dua jam kedepan dibandingkan tampilan prediksi jangka waktu satu jam kedepan. Untuk hasilnya dari kedua perbandingan tersebut masih tetap sama dilihat pada Gambar 4.22 dan Gambar 4.23 dengan data yang suda ada di dalam database.

2023-02-08					2023-02-07				
Jam	Kelembaban	Tekanan Atm	Suhu	Kondisi	Jam	Kelembaban	Tekanan Atm	Suhu	Kondisi
00	24.07%	1016.04 hPa	24.02°C	Berawan	00	22.84%	999.98 hPa	22.94°C	Cerah
01	24.26%	1007.05 hPa	24.26°C	Cerah	01	22.72%	996.73 hPa	22.72°C	Cerah
02	24.82%	1018.28 hPa	24.82°C	HujanY	02	23.43%	1016.7 hPa	23.43°C	Berawan
03	24.58%	1005.07 hPa	24.58°C	Cerah	03	23.02%	1011.07 hPa	23.02°C	HujanY
04	25.25%	1017.25 hPa	25.25°C	HujanY	04	23.87%	997.18 hPa	23.87°C	HujanY
05	25.76%	992.9 hPa	25.76°C	Berawan	05	24.37%	1000.09 hPa	24.37°C	HujanY
06	25.67%	990.81 hPa	25.67°C	Berawan	06	24.02%	1007.15 hPa	24.02°C	HujanY
07	27.5%	1013.05 hPa	27.5°C	Cerah	07	24.37%	1015.21 hPa	24.37°C	Berawan
08	27.88%	1001.13 hPa	27.88°C	Berawan	08	25.58%	1016.12 hPa	25.58°C	HujanY
09	27.05%	999.54 hPa	27.05°C	HujanY	09	26.64%	991.24 hPa	26.64°C	Cerah
10	28.68%	995.57 hPa	28.68°C	Cerah	10	26.23%	998.24 hPa	26.23°C	Cerah
11	29.02%	995.45 hPa	29.02°C	HujanY	11	27%	1011.75 hPa	27°C	Berawan
12	29.22%	994.85 hPa	29.22°C	Cerah	12	27.8%	996.04 hPa	27.8°C	HujanY
13	27.8%	1007.49 hPa	27.8°C	Berawan	13	26.64%	996.45 hPa	26.64°C	Berawan
14	27.56%	1003.23 hPa	27.56°C	Berawan	14	27.06%	1007.36 hPa	27.06°C	Berawan
15	27.49%	1007.34 hPa	27.49°C	HujanY	15	28.37%	1000.91 hPa	28.37°C	HujanY
16	28.28%	1002.62 hPa	28.28°C	Berawan	16	26.81%	991.81 hPa	26.81°C	HujanY
17	28.58%	1001.52 hPa	28.58°C	Cerah	17	27.87%	1011.59 hPa	27.87°C	Berawan
18	27.71%	996.15 hPa	27.71°C	Berawan	18	27.57%	1014.72 hPa	27.57°C	Cerah
19	25.4%	992.96 hPa	25.4°C	Berawan	19	27.63%	1015.14 hPa	27.63°C	Berawan
20	25.88%	998.58 hPa	25.88°C	HujanY	20	25.15%	997.66 hPa	25.15°C	HujanY
21	24.67%	1004.91 hPa	24.67°C	HujanY	21	25.97%	1004.43 hPa	25.97°C	Cerah
22	24.98%	999 hPa	24.98°C	Berawan	22	24.37%	991.34 hPa	24.37°C	Berawan
23	24.91%	1019.09 hPa	24.91°C	Berawan	23	24.61%	1002.55 hPa	24.61°C	Berawan

Gambar 20. Tampilan interface prediksi satu jam



2023-02-08					2023-02-07				
Jam	Kelembaban	Tekanan Atm	Suhu	Kondisi	Jam	Kelembaban	Tekanan Atm	Suhu	Kondisi
00	24.14%	1011.54 hPa	24.14°C	Istimewa	00	22.83%	998.16 hPa	22.81°C	Cerah
02	24.69%	1012.08 hPa	24.69°C	HujanY	02	23.22%	1014.89 hPa	23.22°C	Berawan
04	25.51%	1005.08 hPa	25.51°C	HujanY	04	24.12%	998.63 hPa	24.12°C	HujanY
06	26.59%	1001.93 hPa	26.59°C	Berawan	06	24.19%	1011.17 hPa	24.19°C	HujanY
08	27.45%	1000.33 hPa	27.45°C	Berawan	08	26.11%	1003.73 hPa	26.11°C	HujanY
10	28.84%	995.51 hPa	28.84°C	Cerah	10	26.66%	1005.04 hPa	26.66°C	Cerah
12	28.51%	1000.67 hPa	28.51°C	Cerah	12	27.12%	996.24 hPa	27.12°C	HujanY
14	27.52%	1002.28 hPa	27.52°C	Berawan	14	27.72%	1004.14 hPa	27.72°C	Berawan
16	26.42%	1002.07 hPa	26.42°C	Berawan	16	27.34%	1001.7 hPa	27.34°C	HujanY
18	26.56%	994.56 hPa	26.56°C	Istimewa	18	27.61%	1014.93 hPa	27.61°C	Cerah
20	25.27%	1001.75 hPa	25.27°C	HujanY	20	25.56%	1001.04 hPa	25.56°C	HujanY
22	24.95%	1009.04 hPa	24.95°C	Berawan	22	24.49%	996.95 hPa	24.49°C	Berawan

Gambar 21. Tampilan interface prediksi dua jam.

KESIMPULAN

Pembelajaran mesin memberikan manfaat besar dalam meningkatkan akurasi prediksi cuaca, seperti di Desa Plancungan, di mana cuaca yang berubah-ubah menjadi tantangan bagi petani tembakau untuk menentukan waktu penjemuran hasil panen. Dengan data sebanyak 2.500 dari total 10.000 data yang tersedia, sistem prediksi ini menggunakan 75% data untuk pelatihan dan 25% untuk pengujian, menghasilkan akurasi sebesar 97,16%.

Hasil prediksi mencakup kondisi cuaca saat ini, prediksi satu hingga dua jam ke depan, hingga perkiraan cuaca harian selama 10 hari. Selain itu, perbandingan dengan aplikasi cuaca menunjukkan kesamaan dan perbedaan dalam aspek seperti suhu, tekanan udara, kelembapan, dan kondisi cuaca. Sistem berbasis web ini mempermudah akses bagi petani, membantu mereka lebih percaya diri dalam mengelola proses penjemuran tembakau.

Penelitian berikutnya diharapkan mampu menyajikan informasi yang lebih mendalam dan kompleks mengenai kondisi cuaca di berbagai daerah tertentu. Selain itu, variabel penelitian dapat diperluas untuk memperkaya hasil yang diperoleh. Dari segi tampilan, halaman web sebaiknya dirancang lebih menarik dan intuitif sehingga mudah dipahami oleh masyarakat umum.

DAFTAR RUJUKAN

- [1] S. I. Hadi, E. Ermatita, and N. Chamidah, "Penerapan Fuzzy C-Means dan Fuzzy Sugeno dalam Memprediksi Cuaca," *Inform. J. Ilmu Komput.*, vol. 18, no. 1, pp. 11–22, 2022, doi: <https://doi.org/10.52958/iftk.v17i4.4103>.
- [2] M. Y. R. Rangkuti, M. V. Alfansyuri, and W. Gunawan, "Penerapan Algoritma K-Nearest Neighbor (Knn) Dalam Memprediksi Dan Menghitung Tingkat Akurasi Data Cuaca Di Indonesia," *Hexagon*, vol. 2, no. 2, pp. 11–16, 2021, doi: <https://doi.org/10.36761/hexagon.v2i2.1082>.
- [3] R. Harun, K. C. Pelangi, and Y. Lasena, "Penerapan Data Mining untuk Menentukan Potensi Hujan Harian dengan Menggunakan Algoritma K Nearest Neighbor (KNN)," *J. Manaj. Inform. dan Sist. Inf.*, vol. 3, no. 1, pp. 8–15, 2020, doi: <https://doi.org/10.36595/misi.v3i1.125>.
- [4] M. F. Al Amin, Y. N. Kunang, and S. D. Purnamasari, "Penerapan Model Regression Untuk

- Prediksi Cuaca Wilayah seberang Ulu 1 Palembang,” in *Bina Darma Conference on Computer Science (BDCSS)*, 2019, vol. 1, no. 1, pp. 185–197.
- [5] J. S. Arie, “Implementasi Algoritma KNN Dalam Memprediksi Curah Hujan dan Temperatur Untuk Tanaman Padi,” in *SISITI: Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, 2019, vol. 8, no. 1.
- [6] B. A. Putra, I. Fitri, and R. Nuraini, “PEMBUKA ATAP OTOMATIS PETERNAKAN SAPI BERBASIS MIKROKONTROLER ARDUINO UNO ATMEGA328,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 6, no. 2, pp. 250–258, 2021.
- [7] M. D. Supiannor, F. Fitriyadi, and N. Rosmawanti, “Model Atap Jemuran Gabah Otomatis Berbasis Mikrokontroler Atmega328,” *Progresif J. Ilm. Komput.*, vol. 18, no. 1, pp. 43–54, 2022, doi: <http://dx.doi.org/10.35889/progresif.v18i1.782>.
- [8] A. Lestari and E. Abdulrahman, “Rancang Bangun Modul Raindrop Dan IoT Sebagai Pengendali Penjemur Jagung Marning,” *J. Tek. Elektro Raflesia*, vol. 1, no. 2, pp. 25–31, 2021.
- [9] W. Mira, “Perancangan Prototipe Buka Tutup Atap Otomatis Tempat Penjemuran Kerupuk Berbasis Arduino R3.” Universitas Islam Kalimantan MAB, 2021.
- [10] P. Cunningham and S. J. Delany, “k-Nearest neighbour classifiers: (with Python examples),” *arXiv Prepr. arXiv2004.04523*, 2020.
- [11] P. Cunningham and S. J. Delany, “K-nearest neighbour classifiers-a tutorial,” *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–25, 2021.
- [12] D. Dzulkifli and F. K. Khansa, “Rancang Bangun Perangkap Nyamuk Otomatis Menggunakan Sensor Suhu Dan Kelembaban Dht11 Berbasis Arduino Uno,” *Inov. Fis. Indones.*, vol. 11, no. 2, pp. 28–37, 2022.
- [13] K. P. K. Rianti and Y. Prastyo, “Analisis Penggunaan Sensor Suhu Dan Kelembaban Untuk Monitoring Lingkungan Greenhouse Berbasis Arduino,” *Antivirus J. Ilm. Tek. Inform.*, vol. 16, no. 2, pp. 200–210, 2022.
- [14] M. Khaery, A. H. Pratama, P. Wipradnyana, and A. A. Ngurah, “Perancangan Alat Ukur Tekanan Udara Menggunakan Sensor Barometric Pressure 280 (BMP280) Berbasis Arduino Uno,” *Bul. Fis. Vol.*, vol. 21, no. 1, pp. 14–19, 2020.
- [15] Y. PRATAMA, H. Arief Kusuma, and T. Suhendra, “UJI PERFORMA BMP280: KALIBRASI SENSOR PENGUKURAN TEKANAN UDARA DENGAN REGRESI LINIER BERGANDA.” Universitas Maritim Raja Ali Haji, 2024.