Ibrahimy LIBRARY

KLASIFIKASI CUACA BERDASARKAN DATA SUHU, KELEMBAPAN, DAN CURAH HUJAN MENGGUNAKAN ALGORITMA K-NEIREST NEIGHBOR

SKRIPSI

Oleh :

LEGI OCTA SOFYAN FIRMANDALA

2021503112

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY

SITUBONDO

2025

Ibrahimy Repository

Universitas Ibrahimy

repository.ibrahimy.ac.id

Ibrahimy LIBRARY

KLASIFIKASI CUACA BERDASARKAN DATA SUHU, KELEMBAPAN, DAN CURAH HUJAN MENGGUNAKAN ALGORITMA K-NEIREST NEIGHBOR

SKRIPSI

Oleh :

LEGI OCTA SOFYAN FIRMANDALA

2021503112

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY

SITUBONDO

2025

Universitas Ibrahimy

Ibrahimy Repository

repository.ibrahimy.ac.id

Ibrahimy LIBRARY

HALAMAN JUDUL

KLASIFIKASI CUACA BERDASARKAN DATA SUHU, KELEMBAPAN, DAN CURAH HUJAN MENGGUNAKAN ALGORITMA *K-NEIREST NEIGHBOR*

SKRIPSI

Diajukan untuk Memenuhi Salah Satu Persyaratan dalam Menyelesaikan Program Sarjana (S-1) pada Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Ibrahimy

Oleh :

LEGI OCTA SOFYAN FIRMANDALA

2021503112

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY

SITUBONDO

2025

Ibrahimy Repository

Universitas Ibrahimy

repository.ibrahimy.ac.id

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy

PERNYATAAN KEASLIAN TULISAN

Yang bertanda tangan di bawah ini :

Nama : Legi Octa Sofyan Firmandala

NPM : 2021503112

Prodi : Teknologi Informasi

Menyatakan dengan sebenarnya, bahwa tugas akhir/skripsi ini secara keseluruhan adalah hasil penelitian atau karya saya sendiri, kecuali pada bagian-bagian yang dirujuk sebagai sumber referensi dan disebutkan dalam duftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa tugas ahkir/skripsi ini hasil plagiasi, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Situbondo, 02 JUNI 2025

Legi Octa Sofyan Firmandala

repository.ibrahimy.ac.id

Ibrahimy LIBRARY

LEMBAR PERNYATAAN KESEDIAAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan di bawah ini :

Nama	: Legi Octa Sofyan Firmandala
NPM	: 2021503112
Program Studi	: S-1 Teknologi Informasi
Fakultas	: Fakultas Sains dan Teknologi
Jenis Karya Ilmiah	: Hasil Penelitian

Demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti Non-eksklusif *(Non-exclusive Royalty-Free Right)* kepada Perpustakaan Universitas Ibrahimy atas karya ilmiah saya berupa hasil penelitian yang berjudul :

KLASIFIKASI CUACA BERDASARKAN DATA SUHU, KELEMBAPAN, DAN CURAH HUJAN MENGGUNAKAN ALGORITMA K-NEIREST NEIGHBOR

Dengan Hak Bebas Royalti Non-eksklusif ini Pusat Perpustakaan Universitas Ibrahimy berhak menyimpan, alih media/format, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat untuk dapat dipergunakan sebagaimana mestinya.

Situbondo, 02 Juni 2025

Yang Menyatakan

Legi Octa Sofyan Firmandala

Universitas Ibrahimy
Ibrahimy Repository
repository.ibrahimy.ac.id

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy

PERSETUJUAN PEMBIMBING

Nama : **Legi Octa Sofyan Firmandala**

NPM/NIRM : **2021503112**

Judul : **KLASIFIKASI CUACA BERDASARKAN DATA SUHU, KELEMBAPAN, DAN CURAH HUJAN MENGGUNAKAN ALGORITMA *K-NEIREST NEIGHBOR***

Telah ditelaah dan disetujui oleh pembimbing untuk dujikan pada sidang skripsi/munaqosah

Disetujui oleh:

Pembimbing 1,	Pembimbing 2,
Ahmad Homaidi, M.Kom	**Achmad Baijuri, M.Kom**
NIDN : 0705078901	**NIDN : 0715078902**

repository.ibrahimy.ac.id

Ibrahimy LIBRARY

HALAMAN PENGESAHAN

SKRIPSI

KLASIFIKASI CUACA BERDASARKAN DATA SUHU, KELEMBAPAN, DAN CURAH HUJAN MENGGUNAKAN ALGORITMA K-NEIREST NEIGHBOR

Legi Octa Sofyan Firmandala
2021503112

telah dipertahankan di depan dewan penguji Sidang/Munaqasyah Skripsi pada hari Sabtu, Tanggal 23 agustus 2025 sebagai salah satu syarat memperoleh gelar Sarjana (S.Kom) pada Fakultas Sains dan Teknologi Universitas Ibrahimy.

Tim Penguji,

Ketua Sidang,	Sekretaris Sidang,
Dr. Ach. Khumaidi, M.P.	**Muhammad Khoirroni, S.Kom**
NIDN: 0722049001	NIDN: -
Penguji I,	Penguji II,
Abd. Ghofur, M.Kom.	**Hermanto, M.Kom.**
NIDN: 0711088303	NIDN: 708087807

Mengetahui.

Dekan,

Abd. Ghofur, M.Kom.
NIDN: 0711088303

Universitas Ibrahimy
Ibrahimy Repository
repository.ibrahimy.ac.id

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy

MOTTO

“"Bekerjalah untuk duniamu seolah-olah engkau hidup selamanya, dan beribadahlah untuk akhiratmu seolah-olah engkau mati esok hari." (HR. Baihaqi)”

"Skripsi ini mungkin terasa berat, penuh revisi dan jalan buntu, tapi ingatlah, ini adalah perjalanan yang akan membentuk dirimu. Setiap lembar yang kau tulis adalah jejak dari mimpi yang tak pernah menyerah. Jangan ragu pada langkah kecilmu, karena Allah selalu melihat usaha yang kau perjuangkan. Ketika kau lelah, ingatlah tujuanmu dan doamu, sebab perjalanan ini akan berbuah manis di waktu yang tepat."

-banglegg-

repository.ibrahimy.ac.id

Ibrahimy LIBRARY

PERSEMBAHAN

Dengan segenap rasa syukur yang tiada terkira kepada Allah yang maha segalanya, karya ini saya persembahkan kepada :

1. KHR. Ach. Azaim Ibrahimy, S, Sy M. HI Selaku Pengasuh Pondok Pesantren Salafiyah Syafi'iyah Sukorejo Situbondo beserta sekeluarga besar.
2. Ibu, Ayah, Adik, Almh Kakek, Almh Nenek, dan seluruh keluarga tercinta, terima kasih atas segala perjuangan, doa, serta dukungan tanpa henti yang telah kalian berikan demi masa depan saya. Jerih payah dan kasih sayang kalian akan selalu menjadi kekuatan terbesar dalam hidup saya.
3. Seluruh Dosen Fakultas Sains dan Teknologi yang selalu memberi support dan motivasi.
4. Teman-teman saya di Universitas Ibrahimy Situbondo, khususnya Prodi Teknologi Informasi 2021, terima kasih telah menjadi cahaya dalam perjalanan ini. Kalian yang selalu mengingatkan saat saya melangkah keliru dan hadir memberikan bantuan di saat-saat sulit, adalah anugerah yang tak ternilai dalam hidup saya..
5. Laki-laki sederhana namun terkadang sulit di mengerti isi kepalanya, diri saya sendiri, Legi Octa Sofyan Firmandala yang akrab di panggil Legi, seorang laki-laki berumur 23 tahun. Terima kasih telah bertahan melewati segala kesulitan, meski sering kali terasa seolah dunia menentang. Terima kasih telah memilih untuk terus berusaha, meski terkadang hati ini lelah dan hilang harapan. Skripsi ini bukan hanya tentang ilmu yang didapat, tetapi tentang setiap air mata yang jatuh dalam keheningan malam, tentang rasa cemas yang mengisi setiap langkah, dan tentang perjuangan yang kadang tak terlihat. Di balik tiap kata yang terukir, ada hati yang berjuang untuk tetap berdiri, meski ragu terus datang. Kamu telah berusaha sekuat tenaga, dan meski perjalanan ini tak selalu mudah, kamu tetap memilih untuk terus maju, untuk menunjukkan bahwa tidak ada yang sia-sia dari segala usaha yang telah dilalui.

Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

Ibrahimy LIBRARY

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT. Karena atas Rahman dan Rahim-Nya lah penulis dapat menyelesaikan skripsi ini. Penyusunan skripsi ini tidaklah terlepas dari pihak-pihak yang telah banyak membantu dalam hal segala apapun. Pada kesempatan kali ini, penulis ingin mengucapkan terima kasih kepada :

1. KHR. Ach. Azaim Ibrahimy, S, Sy M. HI selaku Pengasuh Pondok Pesantren Salafiyah Syafi'iyah Sukorejo Situbondo.
2. Bapak KH. Ach. Fadlail, S.H, M.H selaku Rektor Universitas Ibrahimy Situbondo.
3. Bapak Abdul Ghofur, M.Kom selaku Dekan Fakultas Sains dan Teknologi Universitas Ibrahimy Situbondo.
4. Bapak Firman Santoso, M.Kom selaku Ka.Prodi Teknologi Informasi Universitas Ibrahimy Situbondo.
5. Bapak Ahmad Homaidi, M.Kom selaku pembimbing I dan Bapak Ahmad Baijuri, M.Kom selaku pembimbing II yang telah banyak memberikan bimbingan, masukan, dan arahan selama penyusunan skripsi ini.
6. Seluruh dosen dan civitas akademika Universitas Ibrahimy Sukorejo Situbondo yang dengan tulus ikhlas banyak memberi bekal ilmu kepada peneliti selama menempuh pendidikan dari awal hingga akhir semester.

Situbondo, 02 Juli 2025

Penulis,

Legi Octa Sofyan Firmandala

Universitas Ibrahimy
Ibrahimy Repository
repository.ibrahimy.ac.id

DAFTAR ISI

repository.ibrahimy.ac.id

repository.ibrahimy.ac.id

repository.ibrahimy.ac.id

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy

DAFTAR TABEL

repository.ibrahimy.ac.id

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy

DAFTAR GAMBAR

repository.ibrahimy.ac.id

SEGMEN PROGRAM

repository.ibrahimy.ac.id

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy

DAFTAR RUMUS

repository.ibrahimy.ac.id

ABSTRAK

Legi Octa Sofyan Firmandala, 2025. **Klasifikasi Cuaca Berdasarkan Data Suhu, Kelembapan, dan Curah Hujan Menggunakan Algoritma K-Neirest Neighbor.** Skripsi, Program Studi Teknologi Informasi, Universitas Ibrahimy. Pembimbing (I) Ahmad Homaidi, M.Kom., Pembimbing (II) Ahmad Baijuri, M.Kom.

Cuaca merupakan salah satu faktor penting yang memengaruhi berbagai aspek kehidupan manusia, seperti pertanian, transportasi, dan perencanaan kegiatan harian. Untuk memperoleh informasi cuaca yang lebih akurat dan efisien, diperlukan metode klasifikasi yang mampu mengolah data secara otomatis. Penelitian ini bertujuan untuk mengklasifikasikan kondisi cuaca berdasarkan data suhu, kelembapan, dan curah hujan menggunakan algoritma K-Nearest Neighbor (K-NN). Data yang digunakan berasal dari repositori online Kaggle dengan jumlah 13.200 data dan 11 atribut yang relevan dengan kondisi cuaca. Proses klasifikasi dilakukan melalui tahapan pengumpulan data, normalisasi, pemisahan data training dan testing, penerapan algoritma K-NN, serta evaluasi model menggunakan metrik akurasi, presisi, dan recall. Hasil penelitian menunjukkan bahwa nilai K=5 memberikan performa klasifikasi terbaik. Dengan menggunakan RapidMiner sebagai alat bantu, sistem klasifikasi ini mampu mengelompokkan kondisi cuaca ke dalam kategori seperti cerah, berawan, dan hujan secara otomatis. Penelitian ini diharapkan dapat menjadi solusi efektif dalam pengolahan data cuaca serta mendukung pengambilan keputusan di berbagai sektor yang terdampak oleh kondisi atmosfer.

Kata kunci: klasifikasi cuaca, suhu, kelembapan, curah hujan, K-Nearest Neighbor, RapidMiner.

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

ABSTRACT

Legi Octa Sofyan Firmandala, 2024. **Weather Classification Based on Temperature, Humidity and Rainfall Data Using the K-Nearest Neighbor Algorithm**. Thesis, Information Technology Study Program, Ibrahimy University. Supervisor (I) Ahmad Homaidi, M.Kom., Supervisor (II) Ahmad Baijuri, M.Kom.

Weather is a crucial factor that influences various aspects of human life, including agriculture, transportation, and daily activity planning. To obtain more accurate and efficient weather information, a classification method capable of processing data automatically is essential. This study aims to classify weather conditions based on temperature, humidity, and rainfall data using the K-Nearest Neighbor (K-NN) algorithm. The dataset was obtained from an online repository (Kaggle), consisting of 13,200 records and 11 attributes relevant to atmospheric conditions. The classification process includes data collection, normalization, data splitting for training and testing, application of the K-NN algorithm, and model evaluation using accuracy, precision, and recall metrics. The results indicate that using K = 5 yields the best classification performance. With the help of RapidMiner as a data mining tool, the system successfully classifies weather conditions into categories such as sunny, cloudy, and rainy. This research is expected to serve as an effective solution for weather data analysis and support decision-making in various weather-dependent sectors.

Keywords: weather classification, temperature, humidity, rainfall, K-Nearest Neighbor, RapidMiner

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy

BAB I

PENDAHULUAN

1.1 Latar Belakang

Cuaca adalah salah satu faktor penting yang memengaruhi berbagai aspek kehidupan manusia, seperti pertanian, transportasi, dan perencanaan kegiatan sehari-hari[1]. Informasi yang akurat mengenai kondisi cuaca sangat diperlukan. Klasifikasi cuaca berdasarkan data suhu, kelembapan, dan curah hujan merupakan metode untuk mengidentifikasi kondisi cuaca dengan menggunakan informasi dari tiga parameter utama tersebut.

Data suhu, kelembapan, dan curah hujan diperoleh dari stasiun *meteorologi* atau alat pengukur cuaca lainnya yang merekam parameter tersebut dalam periode tertentu[2]. Untuk menganalisis data cuaca yang kompleks dan variatif, digunakan algoritma *K-Nearest Neighbor* (K-NN), sebuah metode dalam *machine learning* untuk klasifikasi data berdasarkan kedekatannya dengan data lain yang telah diketahui klasifikasinya[3]. Algoritma K-NN akan mengklasifikasikan cuaca dengan membandingkan data baru dengan data yang sudah dikategorikan sebelumnya, sehingga menghasilkan identifikasi kondisi cuaca yang lebih akurat dan bermanfaat untuk berbagai aplikasi, seperti perencanaan kegiatan luar ruang, mitigasi bencana, dan peningkatan ketahanan terhadap perubahan cuaca.

Namun, meskipun data cuaca sangat berharga, permasalahan muncul ketika jumlah data yang tersedia sangat besar dan bervariasi, sehingga sulit untuk dianalisis secara manual. Data cuaca biasanya mencakup pengukuran suhu, kelembapan, dan curah hujan yang tercatat selama periode waktu yang panjang dan

repository.ibrahimy.ac.id

tersebar di berbagai lokasi. Mengelola dan menganalisis data sebanyak ini secara *efisien* memerlukan penggunaan teknologi yang dapat menangani volume data yang besar serta mampu menghasilkan informasi yang akurat dan cepat[4]. Oleh karena itu, diperlukan metode yang dapat mengklasifikasikan kondisi cuaca dengan cara yang lebih otomatis dan *efisien.*

Penelitian ini mengatasi tantangan dalam memprediksi dan mengklasifikasikan cuaca, yang merupakan salah satu aspek penting dalam kehidupan sehari-hari. Informasi cuaca yang akurat sangat dibutuhkan untuk berbagai kegiatan, mulai dari pertanian, transportasi, hingga perencanaan bencana. Salah satu metode yang dapat digunakan untuk memprediksi dan mengklasifikasikan cuaca adalah melalui analisis data suhu, kelembapan, dan curah hujan[5].

Penerapan algoritma K-NN dalam klasifikasi cuaca memungkinkan untuk mengidentifikasi kondisi cuaca secara otomatis, sehingga mempermudah pengambilan keputusan yang terkait dengan perencanaan kegiatan atau antisipasi bencana yang dipengaruhi oleh perubahan cuaca. Dengan menggunakan metode ini, diharapkan dapat tercipta sistem yang efisien dalam memprediksi dan mengklasifikasikan cuaca berdasarkan data yang diperoleh, yang pada gilirannya dapat membantu masyarakat dan pihak-pihak terkait dalam mengelola dampak cuaca dengan lebih baik[6].

Manfaat dari penggunaan algoritma K-NN dalam klasifikasi cuaca sangatlah besar. Pertama, dengan menggunakan K-NN, proses klasifikasi cuaca dapat dilakukan secara cepat dan otomatis, yang sebelumnya memerlukan waktu dan

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

upaya manual[7]. Penerapan K-NN dalam klasifikasi cuaca dapat menghasilkan prediksi yang lebih akurat, karena algoritma ini dapat memperhitungkan kedekatan antar data yang memiliki pola serupa. Misalnya, jika suatu daerah memiliki suhu yang tinggi, kelembapan yang tinggi, dan curah hujan yang rendah, algoritma K-NN dapat mengklasifikasikan kondisi cuaca tersebut sebagai cuaca panas atau kering berdasarkan data yang telah ada sebelumnya. Dengan menggunakan data historis, model K-NN dapat terus disempurnakan dan memberikan klasifikasi cuaca yang semakin tepat seiring waktu[8].

Selain itu, sistem klasifikasi cuaca berbasis K-NN juga dapat berfungsi sebagai alat peringatan dini untuk masyarakat dan pihak terkait lainnya. Dengan klasifikasi cuaca yang akurat, sistem ini dapat mengidentifikasi potensi cuaca ekstrem yang dapat terjadi, sehingga masyarakat dapat lebih siap menghadapi risiko cuaca buruk. Misalnya, jika sistem memprediksi adanya hujan deras yang disertai angin kencang, masyarakat dapat diberi peringatan lebih awal untuk mempersiapkan diri dan mengambil tindakan yang diperlukan, seperti menunda aktivitas luar ruangan atau mengungsi jika diperlukan. Hal ini akan sangat membantu dalam mengurangi dampak negatif dari bencana cuaca dan meningkatkan ketahanan masyarakat terhadap perubahan cuaca yang tidak menentu.

Salah satu contoh penelitian dilakukan oleh Moh.Fajrin Sigit Aldy et all. Membahas tentang penggunaan algoritma *K-Nearest Neighbor* (K-NN) untuk klasifikasi cuaca telah menunjukkan potensi yang signifikan, terutama dalam memprediksi curah hujan dan parameter cuaca lainnya. Salah satu studi yang

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

dilakukan di Sulawesi Tengah, misalnya, berhasil memanfaatkan K-NN untuk memprediksi curah hujan dengan akurasi yang cukup baik. Penelitian ini menggunakan data historis yang dikumpulkan dari BMKG untuk mengklasifikasikan curah hujan dan suhu udara, serta mengevaluasi tingkat akurasi pred iksi yang dihasilkan[9].

Penelitian lain yang di lakukan oleh Venia Restreva Danestiara. memprediksi curah hujan. Dalam penelitian ini, data yang digunakan mencakup variabel seperti temperatur, curah hujan, kelembapan, dan kecepatan angin dari tahun 2005 hingga 2016. Dengan menggunakan teknik preprocessing seperti diskritisasi, data cuaca yang telah dipartisi diolah untuk melatih model K-NN. Hasil penelitian ini menunjukkan bahwa metode K-NN dapat menghasilkan prediksi curah hujan yang akurat, dengan perhitungan jarak menggunakan Euclidean Distance dan evaluasi performa menggunakan matriks kebingungannya[10].

Penelitian yang di lakukan oleh Muhammad Yusuf Rizqon Rangkuti et all. menunjukkan bahwa K-NN dapat digunakan untuk menghitung tingkat akurasi prediksi cuaca. Dengan data yang mencakup faktor-faktor seperti fenomena La Nina dan El Nino, algoritma ini dapat membantu para petani dan peternak di Indonesia untuk memprediksi kondisi cuaca yang akan datang dan mengoptimalkan waktu tanam serta kegiatan lain yang bergantung pada cuaca[11].

Selain itu, penelitian yang dilakukan oleh Evania Priyanto et all di Kota Bandung juga membahas penerapan K-NN untuk memprediksi curah hujan, yang mana hasilnya dapat membantu dalam perencanaan pertanian dan sektor-sektor lain yang bergantung pada pola cuaca. Algoritma ini berfungsi dengan menganalisis

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

data suhu, kelembapan, dan curah hujan pada waktu tertentu untuk membuat prediksi yang lebih akurat[12].

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas, maka didapatkan suatu identifikasi masalah yaitu:

a. Data cuaca yang diperoleh dari berbagai sumber, seperti stasiun *meteorologi* dan alat pengukur cuaca lainnya, bersifat kompleks dan tersebar dalam jumlah besar. Hal ini membuat analisis data secara manual menjadi sulit dan tidak efisien.
b. Sistem klasifikasi cuaca berbasis data suhu, kelembapan, dan curah hujan belum banyak diterapkan secara optimal, sehingga masih dibutuhkan penelitian lebih lanjut untuk meningkatkan akurasi dan efisiensinya.

1.3 Rumusan Masalah

Bagaimana penerapan algoritma *K-Nearest Neighbor* (K-NN) dalam klasifikasi cuaca berdasarkan data suhu, kelembapan, dan curah hujan dapat meningkatkan efisiensi dan akurasi dalam menganalisis data cuaca yang kompleks dan tersebar dalam jumlah besar?

1.4 Batasan Masalah

Untuk memudahkan penulisan dalam penyusunan penelitian ini, dan agar pembahasan penelitian ini tidak menyimpang dari apa yang telah dirumuskan, maka penulis membatasi permasalahan sebagai berikut:

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

a. Data cuaca yang digunakan terbatas pada tiga parameter utama, yaitu suhu, kelembapan, dan curah hujan.
b. Algoritma yang digunakan untuk klasifikasi cuaca adalah *K-Nearest Neighbor* (K-NN), tanpa membandingkan atau mengembangkan algoritma *machine learning* lainnya.
c. Penelitian hanya difokuskan pada proses klasifikasi jenis kondisi cuaca (seperti cerah, berawan, hujan), dan tidak mencakup prediksi cuaca jangka panjang atau pemodelan dinamika atmosfer lainnya.

1.5 Tujuan Penelitian

Pada dasarnya segala kegiatan yang dilakukan manusia selalu tertuju pada tujuan dan sasaran yang ingin dicapai. Tujuan dari penyusunan proposal ini sebagai berikut:

a. Menerapkan algoritma *K-Nearest Neighbor* (KNN) untuk mengklasifikasikan kondisi cuaca berdasarkan parameter suhu, kelembapan, dan curah hujan guna menghasilkan klasifikasi yang akurat.
b. Mengevaluasi kinerja algoritma KNN pada dataset cuaca untuk mengetahui tingkat akurasi, presisi, recall, dan f1-score dari hasil klasifikasi yang dihasilkan.

1.6 Manfaat Penelitian

Penulis berharap analisis dan penelitian ini dapat memberikan kontribusi yang dapat diketahui oleh berbagai pihak khususnya penulis, dan pada umumnya oleh semua pihak yang terlibat dalam penyusunan proposal ini, antara lain:

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

a. Meningkatkan Efisiensi Pengolahan Data Cuaca**:** Dengan algoritma K-NN, dapat mengolah data cuaca besar dan bervariasi secara efisien, mengurangi analisis manual, serta mempercepat dan meningkatkan akurasi pengambilan keputusan.

b. Menjadi referensi bagi penelitian selanjutnya yang berfokus pada prediksi dan klasifikasi cuaca berbasis data suhu, kelembapan, dan curah hujan.

1.7 Metode Penelitian

1.7.1 Jenis penelitian

a. Deskriptif Kuantitatif

Menggambarkan data cuaca secara kuantitatif, mengumpulkan data cuaca (suhu, kelembapan, dan curah hujan) dari sumber relevan seperti Kaggle atau Stasiun *Meteorologi*. Data tersebut kemudian akan dianalisis secara kuantitatif dan digunakan untuk menerapkan algoritma *K-Nearest Neighbor* dalam mengklasifikasikan kondisi cuaca secara lebih akurat dan efisien guna mendukung pengambilan keputusan yang tepat

b. Eksperimental

Untuk menguji kinerja algoritma K-NN dalam klasifikasi cuaca berdasarkan data suhu, kelembapan, dan curah hujan melalui eksperimen pada data training dan testing guna mengevaluasi akurasi model.

1.7.2 Tahapan pengumpulan data

Penelitian ini menggunakan teknik Studi Pustaka yaitu mengumpulkan data dari berbagai sumber seperti buku, jurnal, artikel dan situs-situs yang dapat

repository.ibrahimy.ac.id

dipercaya sesuai dengan topik yang digunakan. Teknik pengumpulan data yang digunakan pada penelitian ini sebagai berikut:

a. Kajian Literatur

   Mengacu pada pengumpulan, evaluasi, dan pemahaman terhadap referensi atau sumber-sumber yang ada dalam bidang yang sedang diteliti. https://www.kaggle.com/

b. Pengambilan Data

   Memilih dataset yang sesuai, mengambil data dari berbagai sumber (misalnya, Kaggle dan stasiun *k*), yang mencakup suhu, kelembapan, dan curah hujan dalam periode waktu yang relevan.

1.8 Sistematika Pembahasan

BAB I : PENDAHULUAN

Bab ini mencakup berbagai komponen penting dalam penelitian, yaitu latar belakang, identifikasi masalah, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metode serta jenis penelitian, teknik pengumpulan data, metode pengolahan data, metode pengembangan sistem, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini memuat berbagai literatur yang menjadi landasan pemikiran dalam penelitian ini. Pembahasan mencakup hasil-hasil penelitian terdahulu, landasan teori yang relevan, pemodelan yang digunakan, serta perangkat lunak yang mendukung pelaksanaan penelitian.

BAB III : METODELOGI PENELITIAN

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

Bab ini menjelaskan proses pengumpulan dan pengolahan data yang digunakan dalam penelitian sebelum dianalisis lebih lanjut menggunakan algoritma Data Mining.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil analisis data yang telah diolah, termasuk penjelasan, perhitungan, prediksi yang diperoleh, serta tingkat akurasi dari prediksi tersebut.

BAB V : PENUTUP

Bab terakhir ini memuat kesimpulan dan saran yang dihasilkan dari sistem yang telah diteliti, dirancang, dan diimplementasikan.

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

a. Klasifikasi Cuaca dengan Menggunakan Algoritma *K-Nearest Neighbor*[13].

Merupakan penelitian yang dilakukan oleh D. Dandy pada tahun 2023 di Semarang membahas tentang klasifikasi cuaca berdasarkan data suhu, kelembapan, dan curah hujan menggunakan algoritma *K-Nearest Neighbor* (KNN) dengan tiga kategori cuaca, yaitu cerah (*sunny*), berawan (*cloudy*), dan hujan (*rainy*).

Dataset yang digunakan berjumlah 96.453 baris data dari *website* Kaggle, dibagi menjadi data latih dan data uji dengan rasio 80:20. Algoritma KNN diterapkan dengan parameter nilai K sebesar 3, 5, 7, dan 9, dan kinerjanya dievaluasi menggunakan *confusion matrix* serta *classification report*.

Hasil pengujian menunjukkan bahwa nilai K = 9 memberikan performa terbaik dengan jumlah data terprediksi benar terbanyak, yaitu 13.132 data, serta akurasi tertinggi sebesar 68,073%. Selain itu, *classification report* menunjukkan bahwa semakin besar nilai K, semakin meningkat metrik evaluasi seperti *precision*, *recall*, dan *f1-score* untuk semua kategori cuaca. Hal ini mengindikasikan bahwa nilai K yang lebih besar dapat menghasilkan prediksi yang lebih stabil dan akurat.

b. Klasifikasi Curah Hujan Menggunakan Algoritma *K-Nearest Neighbor* (KNN) di Sulawesi Tengah[9].

repository.ibrahimy.ac.id

Penelitian yang di lakukan oleh Moh.Fajrin Sigit Aldy et all pada tahun 2024 membahas tentang Provinsi Sulawesi Tengah yang terletak dekat garis khatulistiwa mengalami perubahan curah hujan yang tidak menentu, sehingga dapat memicu bencana seperti banjir yang mengganggu aktivitas masyarakat.

Penelitian ini menggunakan metode data mining untuk memprediksi curah hujan dengan dataset BMKG yang dikumpulkan dari 1 Januari 2019 hingga 31 Oktober 2023. Data diklasifikasikan menjadi lima kelas menggunakan algoritma *K-Nearest Neighbor* (KNN).

Hasil penelitian menunjukkan bahwa algoritma KNN dengan nilai K = 23 mencapai akurasi sebesar 83,0%, membuktikan bahwa metode ini memiliki kinerja yang baik dalam klasifikasi dan prediksi curah hujan di Provinsi Sulawesi Tengah.

c. Penerapan Data Mining Untuk Prediksi Perkiraan Hujan dengan Menggunakan Algoritma *K-Nearest Neighbor*[14].

Dibuat Oleh Nursobah Nursobah et all pada tahun 2022 membahas tentang hujan merupakan kondisi di mana tetesan air jatuh dari awan ke bumi dan sangat dinantikan karena dapat membantu profesi seperti petani. Namun, hujan dalam skala besar dapat menghambat aktivitas, terutama kegiatan di luar ruangan, dan bahkan menimbulkan bencana seperti banjir. Oleh karena itu, memperkirakan hujan sangat penting bagi masyarakat untuk mengantisipasi kemungkinan yang dapat terjadi akibat hujan. Kendati demikian, penyampaian informasi prakiraan seringkali tidak merata dan mengalami keterlambatan. Oleh sebab itu, masyarakat diharapkan dapat secara mandiri memprediksi kemungkinan terjadinya hujan. Proses pengolahan data yang benar dan akurat sangat diperlukan, salah satunya dengan

repository.ibrahimy.ac.id

memanfaatkan metode data mining.

Penelitian ini menggunakan algoritma *K-Nearest Neighbor* (KNN) dalam proses klasifikasi. Hasil penelitian menunjukkan bahwa keputusan pengujian data adalah "*NO*," yang mengindikasikan algoritma KNN dan metode data mining mampu membantu menyelesaikan masalah terkait prediksi hujan.

2.2 Landasan Teori

a. *K-Nearest Neighbors*

Algoritma *K-Nearest Neighbor* (KNN) adalah metode klasifikasi data yang menggunakan pendekatan relatif sederhana dibandingkan dengan teknik klasifikasi data lainnya.[15] Merupakan algoritma klasifikasi yang menentukan kelas untuk data baru berdasarkan sejumlah K data terdekat (tetangga) sebagai referensi. *K-Nearest Neighbors* (KNN) adalah algoritma klasifikasi yang bekerja dengan mengidentifikasi kedekatan atau kesamaan data baru terhadap data yang sudah ada. Algoritma ini tidak membangun model eksplisit selama proses pelatihan, melainkan langsung menggunakan data pelatihan untuk menentukan kelas data baru. Proses ini bertujuan untuk mengklasifikasikan objek berdasarkan atribut yang mirip dengan data latih terdekat. K-Nearest Neighbor (KNN) adalah algoritma pembelajaran mesin non-parametrik yang mengklasifikasikan suatu objek berdasarkan jarak terdekat antara objek tersebut dan K objek lain di dalam ruang fitur. Dengan memanfaatkan pendekatan instance-based learning, KNN tidak memerlukan proses pelatihan model eksplisit, tetapi melakukan prediksi langsung dengan membandingkan kesamaan antar data menggunakan metrik seperti Euclidean

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

distance. Kesederhanaan dan fleksibilitasnya membuat KNN banyak digunakan di berbagai bidang.

b. Data Mining

Data mining adalah proses mempelajari dan mengaplikasikan teknik untuk mengungkap pengetahuan berharga dari kumpulan data besar dengan memanfaatkan metode komputasi dan analisis statistik. Proses ini mencakup berbagai tahapan, seperti pemilihan data, prapemrosesan data, pemodelan, evaluasi, hingga interpretasi, untuk menemukan pola, tren, melakukan prediksi, dan mendukung pengambilan keputusan yang lebih efektif. Saat ini, data mining telah diterapkan di berbagai bidang, termasuk bisnis, sains, kesehatan, dan lainnya, untuk mengungkap informasi yang tersembunyi di dalam data[8].

c. *Dataset*

Dataset merupakan sekumpulan data yang berasal dari informasi yang terkumpul dari masa lalu dan dapat dikelola untuk menghasilkan informasi baru dengan menggunakan teknik pembelajaran terawasi *(supervised learning)*[16]. Dalam istilah statistik, *dataset* adalah kumpulan objek yang memiliki atribut atau *variabel* tertentu, di mana setiap objek terdiri dari data dengan sekumpulan atribut atau *variabel*. Objek-objek ini sering disebut dengan istilah lain seperti catatan, titik, vektor, pola, peristiwa, observasi, atau kasus. Sedangkan, baris yang merepresentasikan objek data atau kolom disebut sebagai atribut. Atribut ini kadang-kadang disebut sebagai *variabel*, bidang, fitur, atau dimensi serta berperan penting dalam proses analisis dan

repository.ibrahimy.ac.id

pemodelan data karena menentukan karakteristik dan informasi yang akan dipelajari oleh algoritma.

d. Atribut

Atribut adalah karakteristik atau properti yang dimiliki oleh objek dalam sebuah *dataset*. Atribut ini menggambarkan berbagai aspek dari data yang dikumpulkan dan dapat berupa angka, kategori, atau jenis informasi lainnya. Dalam konteks data mining dan *statistik*, atribut sering kali disebut juga sebagai *variabel*, fitur, atau dimensi, dan digunakan untuk menggambarkan aspek spesifik yang dianalisis dalam proses pemodelan atau klasifikasi data. Atribut ini berfungsi untuk memberikan informasi yang lebih spesifik mengenai objek yang dianalisis, seperti jenis cuaca, tingkat kelembapan, atau kecepatan angin. Dalam analisis data, atribut membantu untuk membedakan dan mengelompokkan objek berdasarkan karakteristik yang dimilikinya, sehingga memudahkan dalam proses pemrosesan dan *interpretasi* data. Atribut juga sering disebut sebagai fitur, *variabel*, kolom, atau atribut rasio[17].

2.3 Perangkat Lunak yang Digunakan

a. *RapidMiner Studio*

RapidMiner adalah *platform* perangkat lunak yang digunakan untuk analisis data, *machine learning*, dan pengolahan data besar. Dengan antarmuka grafis yang ramah pengguna, *RapidMiner* memungkinkan pengguna untuk melakukan *eksplorasi* data, pembersihan data, pemodelan prediktif, dan

repository.ibrahimy.ac.id

evaluasi model tanpa memerlukan keterampilan pemrograman yang mendalam. *Platform* ini mendukung berbagai teknik analisis, termasuk *klasifikasi, regresi, clustering*, dan analisis asosiasi, serta memungkinkan integrasi dengan berbagai sumber data dan alat lainnya untuk mendukung proses analitik yang lebih kompleks. *RapidMiner* menyediakan antarmuka pengguna grafis (GUI) untuk merancang *pipeline analitik*. GUI ini menghasilkan *file* XML yang mendefinisikan proses analitik sesuai dengan keinginan pengguna, yang kemudian dibaca oleh *RapidMiner* untuk menjalankan analisis secara otomatis pada data[18].

RapidMiner adalah sebuah platform komprehensif untuk pengembangan solusi analitik prediktif yang mengintegrasikan teknik statistik, pembelajaran mesin, dan pemrosesan data besar ke dalam satu ekosistem visual yang user-friendly. Dengan dukungan terhadap berbagai algoritma dan kemampuan integrasi dengan bahasa pemrograman seperti Python dan R, RapidMiner memfasilitasi pengambilan keputusan berbasis data secara cepat, akurat, dan efisien.

b. *Python*

Pemrograman tingkat tinggi yang bersifat *interpreted, dynamically typed*, dan multi-paradigm, menjadikannya mudah dipelajari dan digunakan. Dengan sintaks sederhana yang mirip bahasa Inggris, Python mendukung berbagai paradigma pemrograman, termasuk prosedural, berorientasi objek, dan *fungsional*. Keunggulannya terletak pada ekosistem pustaka yang luas,

repository.ibrahimy.ac.id

seperti *NumPy* dan Pandas untuk data *science, TensorFlow* dan *Scikit-learn* untuk *machine learning*, serta *Django* dan *Flask* untuk pengembangan web. *Python* dikembangkan oleh Guido van Rossum dan dirilis pada 1991, dengan fokus pada keterbacaan kode dan *sintaks* yang sederhana. Saat ini, *Python* menjadi salah satu bahasa paling populer, digunakan dalam pengembangan *web*, perangkat lunak, ilmu data, dan *machine learning*[19].

Python adalah bahasa pemrograman serbaguna yang *open-source*, dikembangkan dengan filosofi desain yang menekankan keterbacaan kode, efisiensi pengembangan, dan produktivitas programmer. Dengan dukungan pustaka *(library)* yang sangat luas dan komunitas global yang aktif, *Python* menjadi salah satu fondasi utama dalam revolusi teknologi data, *AI*, dan pengembangan sistem modern.

c. Google Colab

Google Colab *(Collaboratory)* adalah layanan berbasis cloud dari Google yang memungkinkan pengguna menulis dan menjalankan kode Python langsung melalui browser tanpa perlu menginstal perangkat lunak tambahan di komputer lokal[20]. Colab menyediakan lingkungan komputasi yang mendukung penggunaan GPU dan TPU secara gratis, sehingga mempermudah proses komputasi intensif. Selain itu, integrasinya dengan Google Drive memungkinkan pengguna menyimpan, berbagi, dan berkolaborasi secara real-time dalam satu dokumen notebook layaknya Google Docs, namun khusus untuk pemrograman. Kemudahan penggunaan

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

tanpa perlu instalasi lokal, dukungan komputasi tinggi, serta lingkungan kerja yang fleksibel menjadikan Google Colab sebagai alat populer dalam pembelajaran data science, pengembangan kecerdasan buatan, dan penelitian akademik berbasis Python[21].

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Pada dasarnya, metode penelitian adalah serangkaian langkah yang terstruktur untuk mengumpulkan data dengan tujuan tertentu. Ini merupakan suatu pendekatan atau prosedur yang digunakan untuk menyelesaikan suatu masalah atau topik dengan cara yang teliti, terencana, dan sistematis, serta mengikuti prinsip-prinsip ilmiah. Proses penelitian ini bertujuan untuk menemukan fakta, prinsip, atau prosedur yang relevan, serta untuk menguji kebenaran ilmiah suatu informasi yang diperoleh melalui cara-cara yang dapat diulang dan diverifikasi.

Keberhasilan atau kegagalan penelitian sangat bergantung pada pemilihan metode yang tepat. Metode yang sesuai memungkinkan pengumpulan, *analisis*, dan *interpretasi* data secara efektif dan efisien, sementara metode yang salah dapat mengarah pada kesalahan atau hasil yang tidak *valid*. Oleh karena itu, penting untuk memilih metode yang sesuai dengan tujuan penelitian, jenis data, serta keterbatasan waktu dan sumber daya yang ada. Metode yang tepat akan menghasilkan hasil yang objektif dan dapat dipertanggungjawabkan secara ilmiah.

Penelitian ini menggunakan sejumlah 13.200 data yang diperoleh dari Kaggle, dengan pendekatan kuantitatif dan tinjauan literatur online. Data tersebut kemudian diolah dan diklasifikasikan menggunakan algoritma *K-Nearest Neighbors* untuk mengidentifikasi pola dan menghasilkan prediksi yang akurat. Seluruh proses pengolahan data dilakukan melalui tahapan pra-pemrosesan,

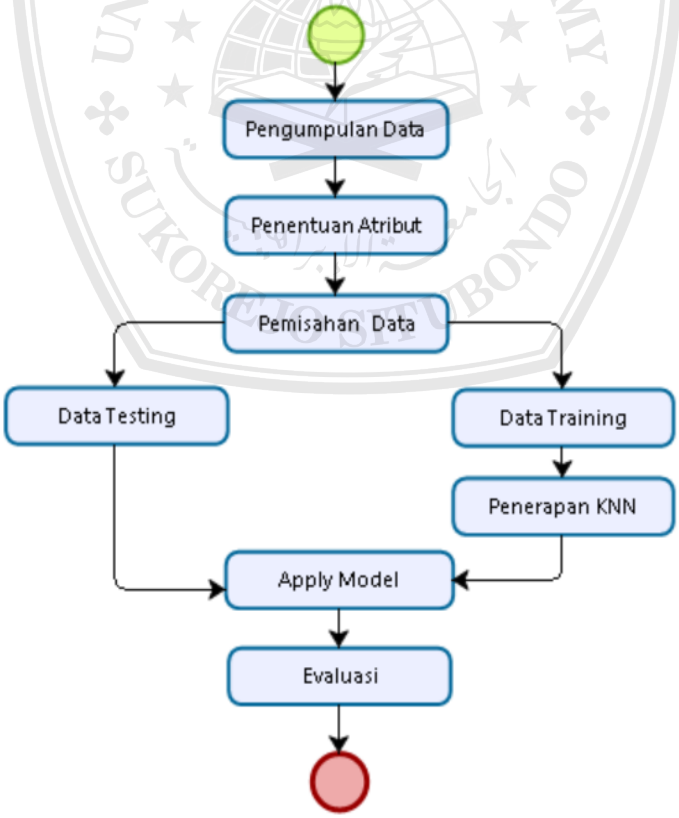
repository.ibrahimy.ac.id

pemilihan parameter K yang optimal, serta evaluasi kinerja model guna memastikan hasil klasifikasi yang tepat.

3.2 Tahapan Penelitian

Sebelum pemodelan data, dilakukan tahap persiapan yang mencakup pengumpulan, transformasi, pembersihan, dan validasi data. Proses ini dapat diulang dalam berbagai urutan sesuai kebutuhan. Langkah utama meliputi pemilihan data yang relevan, menghapus data yang tidak sesuai, serta memastikan data siap untuk proses data mining.

Dalam melakukan penelitian ini dilakukan beberapa tahapan atau langkah penelitian. Sebagaimana gambar dibawah ini :



Gambar 3. 1 Tahapan Penelitian

repository.ibrahimy.ac.id

3.2.1 Pengumpulan Data

Data penelitian diperoleh dari repositori online melalui platform Kaggle, menggunakan dataset *Weather Type Classification*. *Dataset* ini berisi informasi mengenai berbagai kondisi cuaca yang diklasifikasikan berdasarkan parameter seperti suhu, kelembapan, dan curah hujan. Data yang tersedia memungkinkan analisis mendalam menggunakan algoritma KNN untuk mengelompokkan pola cuaca secara lebih akurat. Tautan akses ke dataset tersebut dapat ditemukan di: https://www.kaggle.com/datasets/nikhil7280/weather-type-classification

3.2.2 Penentuan Atribut

Penentuan atribut merupakan langkah penting dalam penelitian ini karena atribut yang dipilih akan memengaruhi akurasi dan efektivitas model klasifikasi cuaca. Atribut adalah *variabel* yang menjadi dasar analisis dan klasifikasi data yang biasanya digunakan dalam penelitian *eksperimental*[17]. Dalam konteks penelitian ini, atribut utama yang digunakan meliputi suhu, kelembapan, dan curah hujan. Ketiga atribut ini dipilih karena secara signifikan mencerminkan kondisi cuaca dan relevan untuk mengidentifikasi pola perubahan cuaca.

Dataset ini berisi 13.200 data yang digunakan untuk klasifikasi cuaca berdasarkan berbagai *parameter atmosfer*. *Dataset* mencakup 11 atribut, yang terdiri dari *variabel input* untuk prediksi cuaca dan satu *variabel target* (*label klasifikasi*). Dalam penelitian ini, dataset yang digunakan terdiri dari 12 atribut**,** yang terbagi menjadi 11 atribut sebagai variabel input (fitur) dan 1 atribut sebagai variabel target (label) yang sudah ditampilkan pada tabel 3.1**.** Atribut-atribut ini dipilih berdasarkan relevansi terhadap kondisi atmosfer yang memengaruhi

repository.ibrahimy.ac.id

klasifikasi cuaca. 11 atribut input meliputi: suhu *(temperature),* kelembapan *(humidity),* curah hujan *(rainfall),* kecepatan angin *(wind speed),* arah angin *(wind direction),* tekanan udara *(pressure),* jarak pandang *(visibility),* titik embun *(dew point),* radiasi matahari *(solar radiation),* tutupan awan *(cloudcover),* dan indeks UV *(UV index)* 1 atribut target adalah *weather condition* yang berfungsi sebagai label klasifikasi, seperti: cerah, berawan, atau hujan. Atribut yang digunakan dalam klasifikasi adalah sebagai berikut:

Tabel 3. 1 Atribut

NO	Nama Atribut	Tipe Data	Deskripsi
1	Temperature	Numerik (float)	Suhu udara dalam derajat Celsius.
2	Humidity	Numerik (float)	Persentase kelembapan udara.
3	Rainfall	Numerik (float)	Curah hujan yang diukur dalam milimeter.
4	Wind Speed	Numerik (float)	Kecepatan angin yang memengaruhi perubahan cuaca.
5	Wind Direction	Kategorial	Arah angin, seperti North, South, East, atau West.
6	Pressure	Numerik (float)	Tekanan udara yang mempengaruhi pembentukan awan dan hujan.
7	Visibility	Numerik (float)	Jarak pandang yang diukur dalam kilometer atau meter.
8	Dew point	Numerik (float)	Titik suhu di mana udara menjadi jenuh dan uap air mulai mengembun.
9	Solar Radiation	Numerik (float)	Intensitas sinar matahari yang diterima permukaan bumi.
10	Cloud Cover	Numerik (float)	Persentase langit yang tertutup oleh awan.
11	UV Index	Numerik (integer)	Indeks radiasi ultraviolet dari sinar matahari.

repository.ibrahimy.ac.id

Tabel 3. 1 (Lanjutan)

12	Weather Condition	Kategorial	Label target klasifikasi : Cerah, Berawan atau Hujan

3.2.3 Pemisahan Data

Pemisahan data, atau yang sering disebut partisi data, merupakan proses membagi kumpulan data menjadi beberapa *subset* dengan tujuan tertentu, seperti untuk pelatihan, *validasi*, dan pengujian[22]. Proses ini melibatkan pembagian *dataset* menjadi dua bagian utama, yaitu data pelatihan *(training data)* dan data pengujian *(testing data)*. Data pelatihan digunakan untuk melatih model sehingga mampu mengenali pola-pola tertentu berdasarkan atribut suhu, kelembapan, dan curah hujan, sedangkan data pengujian digunakan untuk mengevaluasi kinerja model setelah dilatih.

3.2.4 Data *Training*

Pada tahap ini, algoritma *K-Nearest Neighbor* (K-NN) akan mempelajari hubungan antara *variabel* atribut (seperti suhu, kelembapan, dan curah hujan) dan label target (kondisi cuaca). Proses pelatihan mencakup pemberian data dengan label yang sudah diketahui kepada algoritma sehingga model dapat mengenali pola-pola yang ada.

3.2.5 Data *Testing*

Data testing mencakup *subset* data yang telah dipisahkan sebelumnya selama proses pemisahan data. *Data testing* untuk Menganalisis dan memastikan keakuratan model[23]. Terdiri dari atribut *input* (seperti suhu, kelembapan, dan

repository.ibrahimy.ac.id

curah hujan) serta label target yang benar (kondisi cuaca). Model akan membuat prediksi berdasarkan data *input*, dan hasil prediksi tersebut dibandingkan dengan label target sebenarnya untuk menilai akurasi, presisi, *recall*, atau *metrik* evaluasi lainnya.

3.2.6 Penerapan KNN

Penerapan algoritma *K-Nearest Neighbor* (KNN) dilakukan dengan memanfaatkan data training untuk mengklasifikasikan data baru berdasarkan kedekatan atau kemiripan. Proses ini melibatkan perhitungan jarak antara data baru dan seluruh data dalam dataset training, biasanya menggunakan jarak *Euclidean* atau *Manhattan*. Selanjutnya, algoritma memilih sejumlah tetangga terdekat sebanyak nilai K yang telah ditentukan. Kelas dari data baru kemudian ditentukan berdasarkan mayoritas kelas dari tetangga terdekat tersebut, menghasilkan prediksi akhir. Dalam penelitian ini, KNN digunakan untuk mengklasifikasikan kondisi cuaca berdasarkan atribut dengan tujuan menghasilkan prediksi yang akurat.

3.2.7 *Apply Model*

Apply Model adalah tahap penerapan model yang telah dilatih pada data baru untuk melakukan klasifikasi atau prediksi. Setelah melalui proses pengumpulan data, pemisahan data, dan pelatihan menggunakan algoritma *K-Nearest Neighbor* (K-NN), model yang terbentuk digunakan untuk mengolah data uji atau data baru. Pada tahap ini, model akan membandingkan data baru dengan data yang telah dikategorikan sebelumnya berdasarkan kedekatannya dalam ruang fitur. Hasil dari proses ini berupa klasifikasi atau prediksi yang dapat digunakan untuk analisis lebih

repository.ibrahimy.ac.id

lanjut. Tahap ini penting karena menentukan sejauh mana model dapat menggeneralisasi data baru dengan baik sebelum dievaluasi untuk mengukur akurasinya.

3.2.8 Evaluasi

Evaluasi merupakan tahap penting dalam mengukur performa model dalam mengklasifikasikan data. Metode yang digunakan mencakup akurasi, presisi, *recall*, dan *F1-score* untuk menilai keandalan model KNN dalam memprediksi cuaca berdasarkan data testing. Hasil evaluasi ini memastikan bahwa model tidak hanya bekerja baik pada data training, tetapi juga mampu mengklasifikasikan data baru dengan akurasi yang optimal. Selain itu, proses evaluasi juga membantu dalam mengidentifikasi potensi kelemahan model sehingga dapat dilakukan penyesuaian parameter atau teknik pra-pemrosesan untuk meningkatkan kinerjanya.

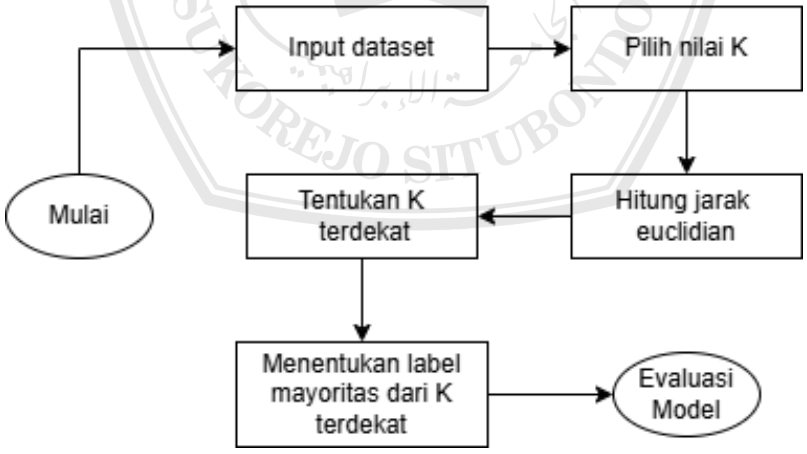
3.3 Perancangan Sistem

3.3.1 Metode algoritma K-NN

Algoritma *K-Nearest Neighbor* (K-NN) merupakan salah satu teknik dalam pembelajaran mesin yang dikategorikan sebagai *non-parametrik* dan termasuk dalam metode *lazy learning.* Sifat *non-parametrik* menunjukkan bahwa algoritma ini tidak memerlukan asumsi awal terhadap distribusi data yang digunakan. Dengan kata lain, K-NN tidak membangun model matematis yang tetap, serta tidak memerlukan parameter khusus untuk menggambarkan struktur data, sehingga dapat digunakan pada berbagai jenis data dengan karakteristik yang berbeda. *K-Nearest Neighbor* (K-NN) merupakan metode klasifikasi yang bekerja dengan menentukan

repository.ibrahimy.ac.id

kelas suatu data berdasarkan mayoritas kelas dari sejumlah tetangga terdekatnya, di mana jarak antar data biasanya diukur menggunakan metode tertentu seperti *Euclidean distance* atau *Manhattan distance*. Sifatnya yang sederhana namun efektif membuat K-NN banyak digunakan dalam berbagai bidang, mulai dari pengenalan pola, analisis data medis, hingga sistem rekomendasi, dengan hasil yang sangat bergantung pada kualitas data dan pemilihan parameter K yang tepat.Metode ini sederhana namun efektif, terutama ketika pola data sulit dipetakan secara linier. Proses klasifikasi ini dilakukan dengan mengidentifikasi K buah data pada dataset latih (*training* data) yang memiliki kemiripan atau kedekatan tertinggi dengan data uji (*testing* data) yang ingin diklasifikasikan. Nilai K tersebut mewakili jumlah tetangga terdekat yang akan digunakan sebagai dasar dalam menentukan label dari data baru[15].



Gambar 3. 2 Alur Perhitungan KNN

Gambar 3. 2 adalah alur perhitungan algoritma *K-Nearest Neighbor* (K-NN): Setelah dapat dataset dimasukkan, langkah pertama yang dilakukan adalah normalisasi data agar seluruh fitur memiliki skala yang seragam, misalnya

repository.ibrahimy.ac.id

menggunakan metode *Min-Max normalization***.** Selanjutnya, tentukan nilai K (misalnya K = 5). Tahap berikutnya adalah menghitung jarak *Euclidean* antara data uji dan seluruh data latih menggunakan rumus yang telah ditentukan.

Rumus jarak *Euclidean*:

$$(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \quad (3.1)$$

Keterangan:

- D(A,B) : jarak Euclidean antara dua titik data A dan B
- Ai : Nilai atribut ke-I dari data A
- Bi : Nilai atribut ke-I dari data B
- n : jumlah atribut (fitur) dalam data

Rumus 3.1 menjelaskan untuk mencari selisih dari data latih (A) ke data uji (B) dan dinyatakan sebagai *Euclidean Distance*. *Euclidean distance* lebih disukai karena sederhana, akurat, dan cocok untuk data numerik kontinu. Dalam algoritma seperti KNN, menunjukkan performa lebih stabil dibanding *Manhattan* dan *Minkowski*, terutama dalam klasifikasi dan *clustering*. Setelah jarak *Euclidean* antara data uji dan seluruh data latih dihitung, langkah berikutnya adalah menentukan K tetangga terdekat dengan cara mengurutkan nilai jarak dari yang terkecil hingga terbesar. Dari K data terdekat tersebut, ditentukan label kelas berdasarkan jumlah kemunculan terbanyak (mayoritas). Proses ini digunakan untuk mengevaluasi kinerja model K-NN dalam melakukan klasifikasi.

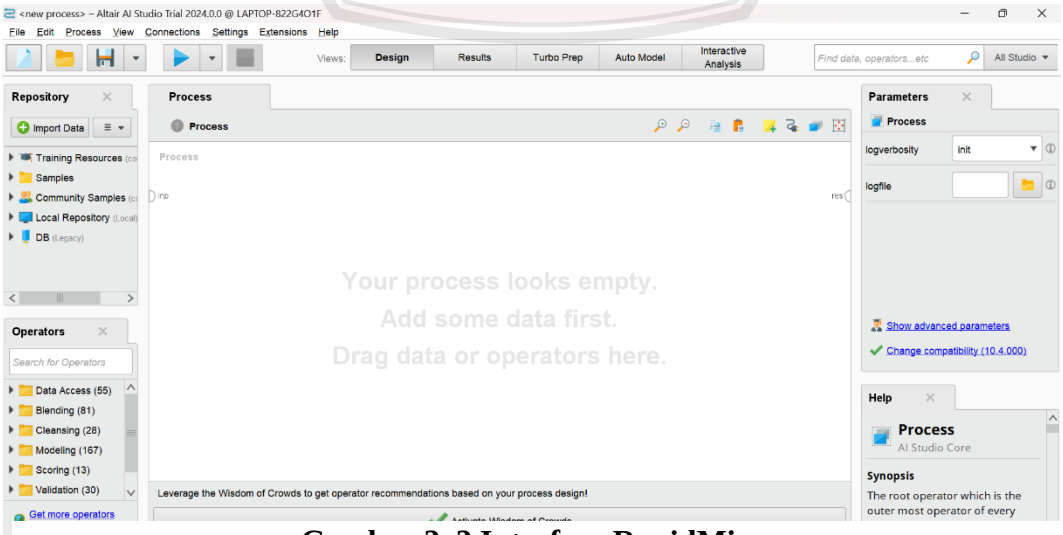
repository.ibrahimy.ac.id

Selanjutnya, model akan menentukan kelas dari data uji berdasarkan hasil evaluasi tersebut. Pada penelitian ini, proses klasifikasi terbagi menjadi tiga label kelas, yaitu Normal, *Suspect*, dan *Pathologic*. Kelas yang memiliki jumlah kemunculan terbanyak dari hasil pembandingan ketiga label tersebut akan dianggap sebagai hasil akhir prediksi oleh algoritma *K-Nearest Neighbor*.

3.4 Implementasi dan pengujian Metode

RapidMiner merupakan perangkat lunak yang direkomendasikan untuk implementasi proyek ini karena mendukung proses data mining secara *efisien.* Aplikasi ini mampu mengekstraksi pola dari dataset berukuran besar dengan memanfaatkan teknik statistik, kecerdasan buatan, dan pengelolaan basis data. RapidMiner menyediakan berbagai model analisis, seperti *K-Nearest Neighbor, Naïve Bayes, Decision Tree,* dan *Neural Network*, yang berfokus pada proses *Knowledge Discovery in Databases* (KDD).

Tampilan antarmuka pengguna dari aplikasi RapidMiner dapat dilihat pada gambar 3.3 dibawah ini.



Gambar 3. 3 Interface RapidMiner

repository.ibrahimy.ac.id

Langkah-langkah implementasi menggunakan RapidMiner mengikuti alur yang telah dijelaskan pada Gambar 3.1 mengenai tahapan penelitian.

Tahap awal dimulai dengan mengimpor data berformat CSV menggunakan fitur Read CSV pada RapidMiner. Selanjutnya, dilakukan pelabelan serta penyesuaian tipe data agar sesuai untuk proses klasifikasi. Setelah itu, data dibagi menjadi data latih dan data uji menggunakan operator Split Data. Untuk mencegah overfitting dan memastikan performa model yang stabil, digunakan operator Cross Validation. Di dalamnya, operator K-NN ditambahkan untuk melatih model, kemudian dikonfigurasi untuk menentukan nilai K yang optimal. Selanjutnya, untuk melakukan evaluasi terhadap model, digunakan operator Performance.

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Penelitian ini dilakukan untuk mengklasifikasikan kondisi cuaca berdasarkan data suhu, kelembapan, dan curah hujan menggunakan algoritma *K-Nearest Neighbors* (K-NN)**.** Dataset yang digunakan diperoleh dari repositori online Kaggle, yang terdiri atas 13.200 data dengan 11 atribut yang merepresentasikan parameter-parameter cuaca yang relevan.

4.2 Perhitungan Manual

4.2.1 Menentukan atribut dan label

Dataset yang digunakan dalam penelitian ini, sebagaimana ditampilkan pada Tabel 4. 1, merupakan kumpulan data cuaca yang diperoleh dari repositori daring *Kaggle*. Dataset ini terdiri dari 13.200 catatan *(record)* yang masing-masing mencakup 11 atribut, yang merepresentasikan berbagai parameter penting kondisi cuaca, seperti suhu udara, kelembapan, curah hujan, tekanan udara, kecepatan angin, serta *indeks UV*. Atribut *'weather_condition'* berfungsi sebagai label kelas (target) yang akan diprediksi berdasarkan kombinasi nilai dari atribut-atribut lainnya. Pemilihan dataset ini dilakukan karena jumlahnya yang cukup besar, variasi datanya yang lengkap, serta keterpercayaan sumbernya, sehingga dapat mendukung analisis klasifikasi cuaca secara akurat. Dataset ini juga memungkinkan penerapan algoritma *K-Nearest Neighbor* (KNN) karena setiap atribut numerik

repository.ibrahimy.ac.id

dapat digunakan untuk perhitungan jarak Euclidean, sehingga mempermudah proses identifikasi.

Tabel 4. 1 Atribut Dataset

Temperature	Humidity	Rainfall	Wind_Speed	Wind_Direction	Pressure	Visibility	Dew_Point	Solar_Radiation	Cloud_Cover	UV_Index	Weather_Condition
30.2	70	5.0	12.4	East	1008	8	24.5	350	20	5	Sunny
28.7	85	12.3	6.8	North	1005	5	23.1	120	90	3	Rainy

Tabel 4. 2 Label Dataset

fh
2
1
1
1
3
3

Tabel 4. 2 menunjukkan atribut label (kelas target) dari dataset cuaca yang digunakan dalam proses pelatihan model *K-Nearest Neighbor* (K-NN). Label ditunjukkan dalam kolom `fh`, yang masing-masing merepresentasikan kategori cuaca: 1 = *sunny*, 2 = *cloudy*, 3 = *rainy*. Nilai-nilai dalam kolom ini merupakan hasil pengklasifikasian berdasarkan atribut cuaca seperti suhu, kelembapan, curah hujan, dan parameter lainnya. Label ini digunakan oleh algoritma K-NN sebagai referensi untuk membandingkan dan menentukan klasifikasi data baru.

Tabel 4. 3 Data Baru yang Akan di Predik

bv	acc	fm	uc	ld	sd	pd	astv	mstv	ptv	mv	ltv	hm	hax	hnp	hnz	hmd	han	hvd	hvr	htd	fh
128	0.002	0	0.02	0.01	0	0	70	0.9	12	1.6	120	2	0	116	130	114	67	115	57	0	?

Tabel 4. 3 memperlihatkan satu baris data baru yang akan diklasifikasikan menggunakan model K-NN. Setiap kolom merepresentasikan nilai dari masing-

repository.ibrahimy.ac.id

masing atribut. Fh label masih kosong dan ditandai dengan **"?"** karena merupakan nilai yang akan diprediksi. Data ini akan dibandingkan dengan data latih menggunakan perhitungan jarak, seperti *Euclidean Distance,* untuk menentukan kelas cuaca yang paling mendekati, yaitu *Sunny***,** *Cloudy,* atau *Rainy***.** Dengan kata lain, sistem akan mencari data yang mirip di dalam dataset, lalu menentukan kelas berdasarkan data terdekat tersebut.

4.2.2 Menentukan nilai K terbaik

Sebelum melakukan proses klasifikasi menggunakan algoritma *K-Nearest Neighbor* (K-NN), penting untuk memastikan bahwa algoritma yang digunakan mendukung jenis metrik evaluasi yang dipilih. Jika algoritma menerapkan metrik yang berbeda, hal tersebut tetap dapat dimanfaatkan untuk memperoleh nilai performa model, seperti akurasi, melalui proses *cross-validation.* Dalam penelitian ini, peneliti menggunakan parameter K = 5**,** karena nilai tersebut memberikan hasil validasi yang paling optimal dan akurasi tertinggi dibandingkan nilai K lainnya.

4.2.3 Menghitung jarak antara data *training* dan data *testing*

Dalam proses perhitungan jarak antara data baru dan data *training*, terdapat berbagai metode yang dapat digunakan, salah satunya adalah *Euclidean Distance.* Metode ini dipilih karena bersifat sederhana, *intuitif,* dan efektif dalam mengukur tingkat kedekatan antar data dalam ruang *multidimensi.* Setelah ditentukan bahwa nilai K = 5, maka langkah selanjutnya adalah menghitung jarak antara setiap data baru terhadap seluruh data dalam dataset training. Hasil perhitungan jarak ini akan digunakan untuk menentukan lima data terdekat yang akan menjadi acuan dalam

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

proses klasifikasi. Sebagaimana yang sudah di cantumkan pada rumus 3.1 Rumus tersebut adalah *jarak Euclidean*, digunakan untuk menghitung jarak lurus (*teorema Pythagoras*) antara dua titik dalam ruang berdimensi.

$$d(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \quad (3.1)$$

Keterangan:

- d(X_1X_2) = jarak antara dua data (X_1 dan X_2).
- n = jumlah atribut/fitur pada data.
- x_{1i} = nilai atribut ke-i dari data pertama.
- x_{2i} = nilai atribut ke-i dari data kedua.
- $(x_{1i} - x_{2i})^2$ = selisih tiap atribut.

Tabel 4. 4 Data Mentah

Data	Suhu (°C)	Kelembapan (%)	Curah hujan (mm)
1	30	80	200
2	25	60	150
3	20	50	50
4	28	75	180
5	22	55	70

Tabel 4.4 adalah tabel data mentah cuaca yang berisi tiga atribut utama, yaitu suhu (°C), kelembapan (%), dan curah hujan (mm) dari lima sampel data. Data ini masih dalam bentuk asli tanpa proses pengolahan, sehingga menjadi dasar sebelum dilakukan tahap normalisasi maupun perhitungan menggunakan algoritma *K-Nearest Neighbor* (KNN).

repository.ibrahimy.ac.id

Berikut adalah perhitungan menggunakan rumus Eulidean Distance:

a. Jarak DT1-DT2

$$d = \sqrt{(30+25)^2 + (80+60)^2 + (200+150)^2}$$

$$= \sqrt{(5)^2 + (20)^2 + (50)^2}$$

$$= \sqrt{25 + 400 + 2500} = \sqrt{2921} = 54.08$$

b. Jarak DT1-DT3

$$d = \sqrt{(30+20)^2 + (80+50)^2 + (200+50)^2}$$

$$= \sqrt{(10)^2 + (30)^2 + (150)^2} =$$

$$= \sqrt{100 + 900 + 22500} = \sqrt{23500} = 153.31$$

c. Jarak DT1-DT4

$$d = \sqrt{(30+28)^2 + (80+75)^2 + (200+180)^2}$$

$$= \sqrt{(2)^2 + (5)^2 + (20)^2}$$

$$= \sqrt{4 + 25 + 400} = \sqrt{429} = 20.71$$

d. Jarak DT1-DT5

$$d = \sqrt{(30+22)^2 + (80+55)^2 + (200+70)^2}$$

$$= \sqrt{(2)^2 + (5)^2 + (20)^2}$$

$$= \sqrt{64 + 625 + 16900} = \sqrt{17589} = 132.63$$

e. Jarak DT2-DT3

$$d = \sqrt{(25+20)^2 + (60+50)^2 + (150+50)^2}$$

$$= \sqrt{(5)^2 + (10)^2 + (100)^2}$$

$$= \sqrt{25 + 100 + 10000} = \sqrt{10125} = 100.62$$

repository.ibrahimy.ac.id

4.2.4 Identifikasi tetangga terdekat

Dalam algoritma *K-Nearest Neighbor* (K-NN)**,** proses klasifikasi dilakukan dengan mencari K data *training* yang memiliki jarak terdekat dengan data baru. Data-data tersebut dibandingkan menggunakan metode *Euclidean Distance***,** yang mengukur seberapa dekat posisi data baru terhadap seluruh data pelatihan dalam ruang berdimensi banyak (berdasarkan fitur/atribut). Setelah dihitung jaraknya, K tetangga terdekat dipilih. Label yang paling banyak muncul di antara mereka akan dijadikan label prediksi untuk data baru.

Tabel 4. 5 Jarak *Euclidean*

No	Data Training	Jarak ke Data Baru	Label
1	DT_001	0.00	*Sunny*
2	DT_004	20.71	*Sunny*
3	DT_002	54.08	*Cloudy*
4	DT_005	132.63	*Sunny*
5	DT_003	153.31	*Rainy*

Berdasarkan data dari K=5 yang dihimpun pada tabel 4.5, teridentifikasi 3 hasil klasifikasi *sunny*, 2 *rainy*, dan 1 *cloudy*. Dengan demikian, maka data baru di beri label *Sunny* dengan hasil klasifikasi terbanyak.

4.2.5 Menghitung nilai *Accuracy*

Setelah proses pemodelan selesai, langkah berikutnya adalah melakukan evaluasi terhadap model untuk mengetahui kinerja algoritma *K-Nearest Neighbor* (K-NN) dalam melakukan klasifikasi. Salah satu metode yang umum digunakan untuk mengevaluasi performa model klasifikasi adalah *Confusion Matrix.*

repository.ibrahimy.ac.id

Confusion matrix merupakan representasi tabular dari hasil klasifikasi yang memperlihatkan perbandingan antara label aktual dan label prediksi. *Matriks* ini terdiri dari empat elemen utama:

a. *Accuracy*

Mengukur seberapa banyak prediksi model yang benar dibandingkan dengan keseluruhan data yang diuji. Semakin tinggi nilai akurasi, maka semakin baik performa model secara umum dalam mengklasifikasikan data. Hal ini menunjukkan kemampuan model dalam mengenali pola yang terdapat pada dataset serta konsistensi hasil prediksi terhadap kondisi sebenarnya. Sebagaimana yang sudah tercantum pada rumus 4.1 adalah rumus akurasi, yaitu ukuran seberapa banyak prediksi model yang benar dibandingkan dengan seluruh data yang diuji.

$$\frac{TP+TN}{TP+TN+FP+FN} \tag{4.1}$$

b. *Recall*

Sensitivity atau (*recall*) mengukur kemampuan model dalam mendeteksi data yang benar-benar positif (atau kelas target tertentu). Sebagaimana yang sudah tercantum pada rumus 4.2 adalah *Recall* (*Sensitivity*), yaitu ukuran seberapa baik model mampu menemukan data positif yang sebenarnya.

$$\frac{TP}{TP+FN} \tag{4.2}$$

c. *Precision*

Presisi adalah seberapa banyak prediksi positif yang benar-benar tepat, tercantum pada rumus 4.3 adalah rumus *precision*, yaitu ukuran seberapa banyak

repository.ibrahimy.ac.id

prediksi positif yang benar dibandingkan dengan semua data yang diprediksi positif oleh model.

$$\frac{TP}{TP+FP} \tag{4.3}$$

d. *F1- score*

F1-Score adalah rata-rata harmonis antara presisi dan recall**,** digunakan untuk menilai performa model secara seimbang ketika ada ketimpangan data yang sudah tercantum pada rumus 4.4 rumus *F1- score.*

$$F1\text{-}Score=2 \text{ x } \frac{Precission\ x\ Recall}{Precission+Recall} \tag{4.4}$$

4.2.6 Normalisasi Data

Tahap normalisasi merupakan proses penting dalam pengolahan data sebelum dilakukan klasifikasi. Normalisasi bertujuan untuk menyamakan skala nilai dari setiap atribut agar tidak ada variabel yang mendominasi perhitungan jarak pada algoritma *K-Nearest Neighbor* (K-NN). Nilai dari masing-masing atribut memiliki rentang yang berbeda, sehingga diperlukan normalisasi agar semua atribut berada pada skala yang sama, yaitu antara 0 hingga 1, tercantum pada rumus 4.5 yaitu rumus *Min-Max Normalization.*

$$X' \frac{X - Xmin}{Xmax - Xmin} \tag{4.5}$$

Keterangan

- X adalah nilai asli,
- Xmin adalah nilai minimum pada atribut,

repository.ibrahimy.ac.id

- Xmax adalah nilai maksimum pada atribut,
- 'X′ adalah nilai hasil normalisasi.

Tabel 4. 6 Data asli Sebelum Normalisasi

Data	Suhu (°C)	Kelembapan (%)	Curah hujan (mm)
1	30	80	200
2	25	60	150
3	20	50	50
4	28	75	180
5	22	55	70

Pada Tabel 4.6 adalah data cuaca sebelum normalisasi, yang memuat atribut suhu, kelembapan, dan curah hujan dalam nilai asli dengan skala berbeda sehingga perlu dinormalisasi agar sebanding.

Cari nilai Min & Max

- Suhu (°C): *min* = 20, *max* = 30
- Kelembapan (%): *min* = 50, *max* = 80
- Curah Hujan (mm): *min* = 50, *max* = 200

Hitung Normalisasi

- Data 1: (30, 80, 200)

$$Suhu = x' \frac{30 - 20}{30 - 20} = \frac{10}{10} = 1.00$$

$$Kelembapan = x' \frac{80 - 50}{30} = \frac{30}{30} = 1.00$$

$$Curah\ Hujan = x' \frac{200 - 50}{150} = \frac{150}{150} = 1.00$$

repository.ibrahimy.ac.id

- Data 2: (25, 60, 100)

$$Suhu = x' \frac{25 - 20}{10} = \frac{5}{10} = 0.50$$

$$Kelembapan = x' \frac{60 - 50}{30} = \frac{10}{30} = 0.33$$

$$Curah\ Hujan = x' \frac{100 - 50}{150} = \frac{50}{150} = 0.33$$

- Data 3: (20, 50, 50)

$$Suhu = x' \frac{20 - 20}{10} = 10 = 0.00$$

$$Kelembapan = x' \frac{50 - 50}{30} = 30 = 0.00$$

$$Curah\ Hujan = x' \frac{50 - 50}{150} = 150 = 0.00$$

- Data 4: (28, 75, 180)

$$Suhu = x' \frac{28 - 20}{10} = \frac{8}{10} = 0.80$$

$$Kelembapan = x' \frac{75 - 50}{30} = \frac{25}{30} = 0.83$$

$$Curah\ Hujan = x' \frac{180 - 50}{150} = \frac{130}{150} = 0.87$$

- Data 5: (22, 55, 70)

$$Suhu = x' \frac{22 - 20}{10} = \frac{2}{10} = 0.20$$

$$Kelembapan = x' \frac{55 - 50}{30} = 30 = 0.17$$

repository.ibrahimy.ac.id

$$Curah\ Hujan = x' \frac{70 - 50}{150} = 150 = 0.13$$

Tabel 4. 7 Data Setelah Normalisasi

Data	Suhu (°C)	Kelembapan (%)	Curah hujan (mm)
1	1.00	1.00	1.00
2	0.50	0.33	0.33
3	0.00	0.00	0.00
4	0.80	0.83	0.87
5	0.20	0.17	0,13

Tabel 4.7 adalah tabel data cuaca yang telah dinormalisasi menggunakan metode *Min-Max*, sehingga seluruh atribut suhu, kelembapan, dan curah hujan berada dalam rentang 0 hingga 1. Proses normalisasi ini dilakukan untuk menyamakan skala data agar tidak ada atribut yang lebih dominan dalam perhitungan jarak pada algoritma *K-Nearest Neighbor* (KNN).

4.3 Implementasi Program

a. Upload Dataset

Segmen Program 4. 1 *Import File* Dataset

```
from google.colab import files
uploaded = files.upload()  # Upload
'weather_classification_data.csv'
```

Segmen Program 4.1 merupakan kode *from google.colab import files* dan *files.upload*() digunakan di *Google Colab* untuk mengunggah *file* dari komputer ke lingkungan kerja *Colab*. Saat kode dijalankan, akan muncul tombol untuk memilih *file* yang ingin diunggah. Setelah file dipilih, *file*

repository.ibrahimy.ac.id

tersebut bisa langsung digunakan dalam pemrosesan data, misalnya untuk membaca dataset dalam format CSV.

b. Pengujian Model

Segmen Program 4. 2 Pengujian Model KNN

```
import pandas as pd
import numpy as np
6import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.neighbors import KneighborsClassifier
from sklearn.metrics import confusion_matrix,
classification_report, accuracy_score,
precision_recall_fscore_support
```

Segmen Program 4.2 berfungsi memanggil pustaka/*library* yang digunakan untuk membangun model klasifikasi cuaca dengan algoritma *K-Nearest Neighbor* (K-NN). Beberapa *library* seperti *pandas* dan *numpy* digunakan untuk mengolah data, sedangkan *matplotlib* dan *seaborn* dipakai untuk membuat grafik. Kode ini juga mempersiapkan fungsi pembagian data menjadi data latih dan data uji, lalu mengubah data kategori menjadi angka agar bisa diproses. Model K-NN digunakan untuk memprediksi kelas cuaca, dan hasilnya dievaluasi menggunakan metrik seperti akurasi, presisi, *recall*, *dan f1-score*.

c. Menyimpan Dataset

Segmen Program 4. 3 Menyimpan Data ke Variabel df

```
df = pd.read_csv('weather_classification_data.csv')
df.head()
```

Segmen Program 4.3 berfungsi untuk membaca data dari *file* CSV bernama *weather_classification_data.csv* dan menyimpannya ke dalam variabel df.

repository.ibrahimy.ac.id

Setelah itu, *df.head*() digunakan untuk menampilkan 5 baris pertama dari data agar kita bisa melihat isi dan bentuk data yang akan digunakan.

d. Mengubah Data

Segmen Program 4. 4 Mengubah Data Menjadi Angka

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Dataset sederhana (contoh penelitian cuaca)
data = {
    'Cloud Cover': ['Clear', 'Cloudy', 'Overcast',
'Clear', 'Cloudy'],
    'Season': ['Summer', 'Winter', 'Spring',
'Autumn', 'Summer'],
    'Weather Type': ['Rainy', 'Cloudy', 'Sunny',
'Sunny', 'Rainy']
}

df = pd.DataFrame(data)
print("=== Data Asli ===")
print(df)

# Label encoding untuk kolom kategori
df_encoded = df.copy()
categorical_cols = ['Cloud Cover', 'Season']
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df[col])
    label_encoders[col] = le
df_encoded = df.copy()
categorical_cols = ['Cloud Cover', 'Season']
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df[col])
    label_encoders[col] = le

# Label encoding untuk target (Weather Type)
target_le = LabelEncoder()
df_encoded['Weather Type'] =
target_le.fit_transform(df['Weather Type'])

print("\n=== Data Setelah Label Encoding ===")
print(df_encoded)
# Pisahkan fitur dan target
X = df_encoded[['Cloud Cover', 'Season']]  # hanya
fitur ini
```

repository.ibrahimy.ac.id

Segmen Program 4. 4 (Lanjutan)

```
        y = df_encoded['Weather Type']                # target #
        Split data: 80% training, 20% testing
        X_train, X_test, y_train, y_test = train_test_split(
            X, y, test_size=0.2, random_state=42
        )

        print("\n=== Data Training (X_train) ===")
        print(X_train)

        print("\n=== Data Testing (X_test) ===")
        print(X_test)

        print("\n=== Label Training (y_train) ===")
        print(y_train)

        print("\n=== Label Testing (y_test) ===")
print(y_test)
```

Segmen Program 4.4 berfungsi untuk melakukan *preprocessing* data cuaca sederhana. Pertama, data cuaca dibuat dalam bentuk *dictionary* lalu dikonversi menjadi *DataFrame* dengan tiga atribut, yaitu *Cloud Cover*, *Season*, dan *Weather Type*. Selanjutnya dilakukan *Label Encoding* untuk mengubah nilai kategori pada kolom *Cloud Cover* dan *Season* menjadi angka agar dapat diproses oleh algoritma *machine learning* sebagaimana yang tercantum pada gambar 4.1 *Output Encoding* Data, sedangkan *Weather Type* sebagai target juga diencoding dengan *LabelEncoder*.

```
=== Data Asli ===
  Cloud Cover  Season Weather Type
0       Clear  Summer        Rainy
1      Cloudy  Winter       Cloudy
2    Overcast  Spring        Sunny
3       Clear  Autumn        Sunny
4      Cloudy  Summer        Rainy

=== Data Setelah Label Encoding ===
   Cloud Cover  Season  Weather Type
0            0       2             1
1            1       3             0
2            2       1             2
3            0       0             2
4            1       2             1
```

Gambar 4. 1 *Output Encoding* Data

repository.ibrahimy.ac.id

```
=== Data Training (X_train) ===
   Cloud Cover  Season
4            1       2
2            2       1
0            0       2
3            0       0

=== Data Testing (X_test) ===
   Cloud Cover  Season
1            1       3

=== Label Training (y_train) ===
4    1
2    2
0    1
3    2
Name: Weather Type, dtype: int64

=== Label Testing (y_test) ===
1    0
```

Gambar 4. 2 *Output Split* Data

Gambar 4.2 adalah proses *split* data menggunakan *train_test_split*, di mana 80% data dijadikan *training* set dan 20% sisanya sebagai *testing* set. Terakhir, hasil pemisahan data ditampilkan berupa data *training* dan testing untuk fitur maupun labelnya, sehingga dataset siap digunakan dalam proses klasifikasi.

e. Pemodelan KNN

Segmen Program 4. 5 Pemodelan KNN K = 5

```
import pandas as pd
import numpy as np
from collections import Counter

# ==============================
# Data Mentah
# ==============================
data = {
    "Data Training": ["DT_001", "DT_002", "DT_003",
"DT_004", "DT_005"],
    "Suhu": [30, 25, 20, 28, 22],
    "Kelembapan": [80, 60, 50, 75, 55],
    "Curah Hujan": [200, 150, 50, 180, 70],
    "Label": ["Sunny", "Cloudy", "Rainy", "Sunny",
"Sunny"]
}

df = pd.DataFrame(data)
```

repository.ibrahimy.ac.id

Segmen Program 4. 5 (Lanjutan)

```
        # ==============================
        # Data Baru (contoh: sama dengan DT_001)
        # ==============================
        data_baru = np.array([30, 80, 200])

        # ==============================
        # Fungsi Euclidean Distance
        # ==============================
        def euclidean_distance(row, data_baru):
            return np.sqrt(np.sum((row - data_baru) ** 2))

        # Hitung jarak untuk semua data training
        df["Jarak ke Data Baru"] = df.iloc[:,
        1:4].apply(lambda row: euclidean_distance(row.values,
        data_baru), axis=1)

        # Urutkan berdasarkan jarak dan ambil K=5
        k = 5
        df_sorted = df.sort_values(by="Jarak ke Data
        Baru").head(k).reset_index(drop=True)

        # Voting mayoritas
        labels = df_sorted["Label"].tolist()
        prediksi = Counter(labels).most_common(1)[0][0]

        # ==============================
        # Hasil
        # ==============================
        df_hasil = df_sorted[["Data Training", "Jarak ke Data
        Baru", "Label"]]
        df_hasil.index = df_hasil.index + 1  # nomor mulai
        dari 1
        print("Hasil KNN (K=5):\n")
        print(df_hasil.to_string())
print("\nPrediksi akhir untuk data baru adalah:", prediksi)
```

Segmen Program 4.5 berfungsi untuk implementasi algoritma *K-Nearest Neighbor* (KNN) dengan K=5, data mentah berisi lima data training dengan atribut Suhu, Kelembapan, dan Curah Hujan beserta label cuaca disimpan dalam bentuk *DataFrame* agar mudah diolah. Selanjutnya ditentukan data baru (dalam contoh sama dengan DT_001), kemudian dibuat fungsi `euclidean_distance` untuk menghitung jarak *Euclidean* antara setiap data training dan data baru. Perhitungan jarak dilakukan dengan mengkuadratkan

repository.ibrahimy.ac.id

selisih tiap atribut, menjumlahkannya, lalu diakarkan. Hasil jarak ini disimpan pada kolom "Jarak ke Data Baru"**,** kemudian data training diurutkan berdasarkan jarak terkecil dan dipilih sebanyak 5 tetangga terdekat (K=5)**.** Dari kelima tetangga tersebut dilakukan voting mayoritas untuk menentukan label prediksi, misalnya jika label terbanyak adalah *Sunny* maka hasil klasifikasi juga *Sunny*. Terakhir, hasil perhitungan ditampilkan dalam bentuk tabel berisi nama data training, jarak ke data baru, serta label, kemudian diikuti dengan *output* prediksi akhir yang memberikan gambaran jenis cuaca berdasarkan data uji sebagaimana yang sudah ditampilkan pada gambar 4.3 Hasil KNN K = 5.

```
Hasil KNN (K=5):

  Data Training  Jarak ke Data Baru   Label
1        DT_001           0.000000   Sunny
2        DT_004          20.712315   Sunny
3        DT_002          54.083269  Cloudy
4        DT_005         132.623527   Sunny
5        DT_003         153.297097   Rainy

Prediksi akhir untuk data baru adalah: Sunny
```

Gambar 4. 3 Hasil KNN K = 5

f. Pelebelan *encoding Weather type*

Segmen Program 4. 6 Label Encoder

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Contoh dataset sederhana sesuai penelitianmu
data = {
    'Suhu (°C)': [30, 25, 10, 28, 15, 35, 5, 20],
    'Kelembapan (%)': [80, 70, 90, 75, 60, 50, 95,
65],
    'Curah Hujan (mm)': [200, 150, 50, 180, 70, 20,
100, 60],
    'Weather Type': ['Rainy', 'Cloudy', 'Snowy',
'Rainy', 'Sunny', 'Sunny', 'Snowy', 'Cloudy']   }
```

repository.ibrahimy.ac.id

Segmen Program 4. 6 (Lanjutan)

```
    # Buat DataFrame
    df = pd.DataFrame(data)
    # Label Encoding untuk kolom target 'Weather Type'
    le = LabelEncoder()
    df['Weather Encoded'] = le.fit_transform(df['Weather
    Type'])
    # Tampilkan mapping label
    print("=== Mapping Label Cuaca ===")
    for i, label in enumerate(le.classes_):
        print(f"{label} -> {i}")
    # Tampilkan data hasil encoding
    print("\n=== Data dengan Label Encoding ===")
    print(df)
```

Segmen Program 4.6 digunakan untuk mengubah data cuaca yang berbentuk teks seperti *Cloudy*, *Rainy*, *Snowy*, dan *Sunny* menjadi angka agar bisa diproses oleh algoritma *machine learning.* Proses ini disebut *label encoding.* Dengan *LabelEncoder(),* setiap jenis cuaca diberi nomor, *Cloudy* → 0, *Rainy* → 1, *Snowy* → 2, *Sunny* → 3. Hasil konversi ditampilkan dalam tabel baru pada kolom *Weather Encoded* yang telah ditampilkan pada gambar 4.4 *Output* Label *Encoder*. Jadi, fungsi kode ini adalah menyederhanakan data kategori menjadi angka supaya bisa digunakan dalam pelatihan model.

```
=== Mapping Label Cuaca ===
Cloudy -> 0
Rainy -> 1
Snowy -> 2
Sunny -> 3

=== Data dengan Label Encoding ===
   Suhu (°C)  Kelembapan (%)  Curah Hujan (mm) Weather Type  Weather Encoded
0         30              80              200        Rainy                1
1         25              70              150       Cloudy                0
2         10              90               50        Snowy                2
3         28              75              180        Rainy                1
4         15              60               70        Sunny                3
5         35              50               20        Sunny                3
6          5              95              100        Snowy                2
7         20              65               60       Cloudy                0
```

Gambar 4. 4 *Output* Label *Encoder*

repository.ibrahimy.ac.id

g. Normalisai Data Awal ke Data Baru

Segmen Program 4. 7 Normalisasi

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

# === Data Sebelum Normalisasi ===
data = {
    'Suhu (°C)': [30, 25, 20, 28, 22],
    'Kelembapan (%)': [80, 60, 50, 75, 55],
    'Curah Hujan (mm)': [200, 150, 50, 180, 70]
}

df = pd.DataFrame(data)
print("=== Data Sebelum Normalisasi ===")
print(df)
# === Normalisasi Min-Max ===
scaler = MinMaxScaler()
df_normalized =
pd.DataFrame(scaler.fit_transform(df),
columns=df.columns)

print("\n=== Data Sesudah Normalisasi (0 - 1) ===")
print(df_normalized.round(2))  # dibulatkan 2 angka
biar lebih sederhana
```

Segmen Program 4.7 digunakan untuk melakukan normalisasi data cuaca menggunakan metode *Min-Max Scaling*. Pertama, dibuat sebuah dataset sederhana dengan tiga atribut yaitu Suhu (°C), Kelembapan (%), dan Curah Hujan (mm) yang kemudian ditampilkan sebagai data awal.

```
=== Data Sebelum Normalisasi ===
   Suhu (°C)  Kelembapan (%)  Curah Hujan (mm)
0         30              80               200
1         25              60               100
2         20              50                50
3         28              75               180
4         22              55                70

=== Data Sesudah Normalisasi (0 - 1) ===
   Suhu (°C)  Kelembapan (%)  Curah Hujan (mm)
0        1.0            1.00              1.00
1        0.5            0.33              0.33
2        0.0            0.00              0.00
3        0.8            0.83              0.87
4        0.2            0.17              0.13
```

Gambar 4. 5 *Output* Normalisasi Data

repository.ibrahimy.ac.id

Selanjutnya, digunakan *MinMaxScaler* dari *scikit-learn* untuk mengubah setiap nilai pada dataset ke dalam rentang 0 sampai 1 berdasarkan nilai minimum dan maksimum dari masing-masing kolom. Hasil normalisasi kemudian ditampilkan dalam bentuk *DataFrame* baru dengan nilai yang sudah diskalakan yang tertampilkan pada gambar 4.5 *Output* Normalisasi Data, kemudian dibulatkan hingga dua angka di belakang koma agar lebih mudah dibaca.

h. Visualisasi *Confusion Matrix*

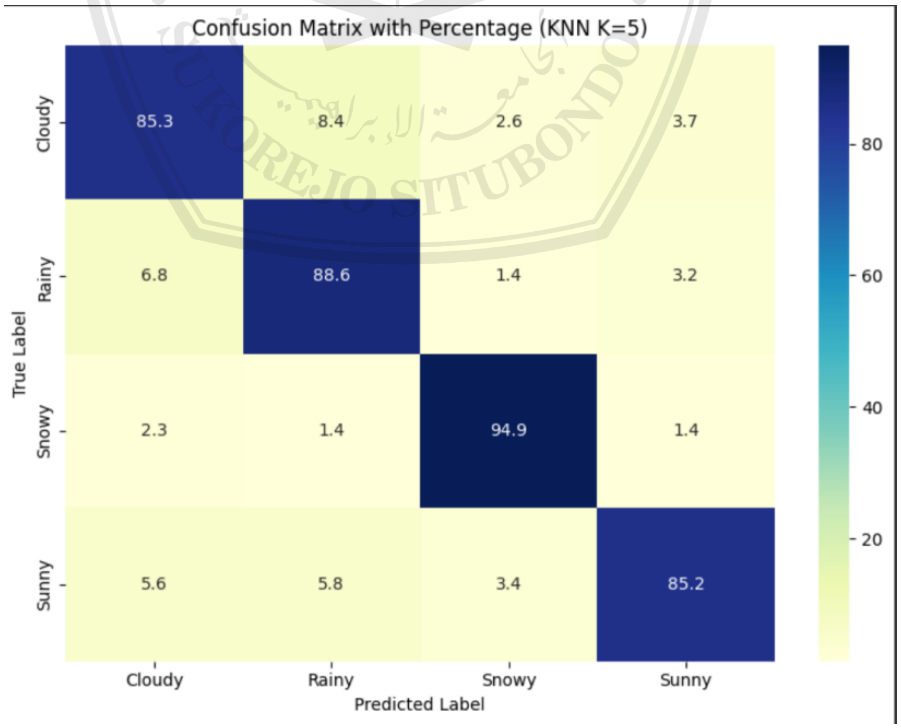
Segmen Program 4. 8 *Confusion Matrix*

```
conf_matrix = confusion_matrix(y_test, y_pred)
labels = target_le.classes_
percent_matrix = (conf_matrix /
conf_matrix.sum(axis=1, keepdims=True)) * 100
plt.figure(figsize=(8, 6))
fmt='.1f', cmap='YlGnBu', xticklabels=labels,
yticklabels=labels)
plt.title('Confusion Matrix with Percentage (KNN
K=5)')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.tight_layout()
plt.show()
```

Segmen Program 4.8 digunakan untuk menampilkan *confusion matrix* dalam bentuk persentase untuk model *K-Nearest Neighbor* (KNN) dengan nilai K=5**.** Proses dimulai dengan pemanggilan fungsi `confusion_matrix(y_test, y_pred)` untuk menghitung jumlah prediksi benar dan salah pada setiap kelas berdasarkan hasil uji model. *Variabel* `labels` menyimpan daftar nama kelas yang diambil dari `target_le.classes` untuk mempermudah pelabelan sumbu grafik. Matriks hasil perhitungan kemudian diubah menjadi persentase dengan cara membagi

repository.ibrahimy.ac.id

setiap elemen pada baris dengan total nilai di baris tersebut (`conf_matrix.sum(axis=1, keepdims=True)`) lalu dikalikan 100, sehingga perbandingan tiap kelas dapat terlihat secara proporsional. Hasil persentase ini divisualisasikan menggunakan fungsi `sns.heatmap` yang menghasilkan *heatmap* berwarna biru-hijau (`cmap='YlGnBu'`) dengan nilai persentase yang ditampilkan langsung di dalam setiap sel (`annot=True, fmt='.1f'`). Sumbu X merepresentasikan label prediksi, sedangkan sumbu Y merepresentasikan label sebenarnya, dapat dikategorikan data sebagai berikut, yaitu jumlah prediksi benar (*True Positive*) dan salah (*False Positive* maupun *False Negative)* untuk setiap kelas dapat terlihat dengan jelas, maka dapat dikategorikan data yang di tampilkan pada gambar 4.6 *Confusion Matrix.*



Gambar 4. 6 *Confusion Matrix*

repository.ibrahimy.ac.id

i. Menghitung Akurasi

Tabel 4. 8 Pemetaan Data

Class	TP	FP	FN
Cloudy	85.3	14.7	14.7
Rainy	88.6	15.6	11.4
Snowy	94.9	7.4	5.1
Sunny	85.2	8.3	14.8

Keterangan:

- TP: Prediksi positif yang benar.
- TN: Prediksi negatif yang benar.
- FP: Prediksi positif tapi salah.
- FN: Prediksi negatif tapi salah

Tabel 4.8 adalah tabel pemetaan data hasil perhitungan *confusion matrix* yang menunjukkan nilai *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN) dari masing-masing kelas cuaca, yaitu *Cloudy*, *Rainy*, *Snowy*, dan *Sunny*. Nilai tersebut menggambarkan tingkat keberhasilan maupun kesalahan model *K-Nearest Neighbor* (KNN) dalam melakukan klasifikasi, sehingga dapat digunakan sebagai dasar untuk menghitung metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*.

Segmen Program 4. 9 Hitung Akurasi

```
# 📊 Menghitung Akurasi Berdasarkan Confusion Matrix
Persentase KNN (K=5)

# Nilai True Positive (TP) dari diagonal confusion
matrix
tp_cloudy = 85.3
tp_rainy = 88.6
tp_snowy = 94.9
```

repository.ibrahimy.ac.id

Segmen Program 4. 9 (Lanjutan)

```
tp_sunny = 85.2

# Menjumlahkan semua nilai TP
total_tp = tp_cloudy + tp_rainy + tp_snowy + tp_sunny

# Karena tiap baris merepresentasikan 100%, dan ada 4
kelas
total_data_percentage = 100 * 4

# 🔍 Rumus akurasi: (jumlah TP / total data) x 100
accuracy = (total_tp / total_data_percentage) * 100
# Menampilkan hasil
print("=== Perhitungan Akurasi Klasifikasi Cuaca (KNN
K=5) ===")
print(f"TP Cloudy : {tp_cloudy}%")
print(f"TP Rainy  : {tp_rainy}%")
print(f"TP Snowy  : {tp_snowy}%")
print(f"TP Sunny  : {tp_sunny}%")
print(f"Total TP  : {total_tp}%")
print(f"Akurasi   : {accuracy:.2f}%")
```

Segmen Program 4.9 berfungsi menghitung akurasi model KNN (K=5) berdasarkan *confusion matrix* dalam bentuk persentase, di mana nilai *True Positive* (TP) untuk tiap kelas cuaca (*Cloudy*, *Rainy*, *Snowy*, *Sunny*) diambil dari nilai diagonal *matriks*. Jumlah seluruh TP disimpan dalam variabel total_tp, sedangkan total_data_*percentage* bernilai 400 karena terdapat empat kelas yang masing-masing mewakili 100%. Akurasi dihitung menggunakan rumus (Total TP/Total Data)×100 (Total TP/Total Data)×100.

```
=== Perhitungan Akurasi Klasifikasi Cuaca (KNN K=5) ===
TP Cloudy : 85.3%
TP Rainy  : 88.6%
TP Snowy  : 94.9%
TP Sunny  : 85.2%
Total TP  : 353.99999999999994%
Akurasi   : 88.50%
```

Gambar 4. 7 *Output* Akurasi KNN K = 5

repository.ibrahimy.ac.id

Seperti yang ditampilkan pada gambar 4.7 adalah *Output* Akurasi KNN K = 5. Program menampilkan nilai TP tiap kelas, total TP keseluruhan, dan persentase akurasi akhir. Akurasi menunjukkan proporsi prediksi yang tepat dibandingkan dengan seluruh jumlah data yang diuji. Ukuran ini paling sesuai digunakan ketika distribusi setiap kelas relatif seimbang. Untuk mencari nilai akurasi maka perlu menggunakan rumus yang sudah ditampilkan pada rumus 4.1 yang diamana untuk mengukur seberapa banyak prediksi model yang benar dibandingkan dengan seluruh jumlah data uji.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Perhitungan akurasi *K-Nearest Neighbor* :

TP = jumlah diagonal utama = 85.3 + 88.6 + 94.9 + 85.2 = 354.0

Total data = semua nilai dijumlahkan = 100 × 4 = 400 (karena setiap baris mewakili 100%)

$$\text{Akurasi} = \frac{85.3 + 88.6 + 94.9 + 85.2}{13.200} = \frac{354.0}{400} = \mathbf{0.885}$$

Jadi akurasinya = 88,50%

j. Menghitung Nilai *Precision, F1 Score* dan *Recall*

Segmen Program 4. 10 Nilai

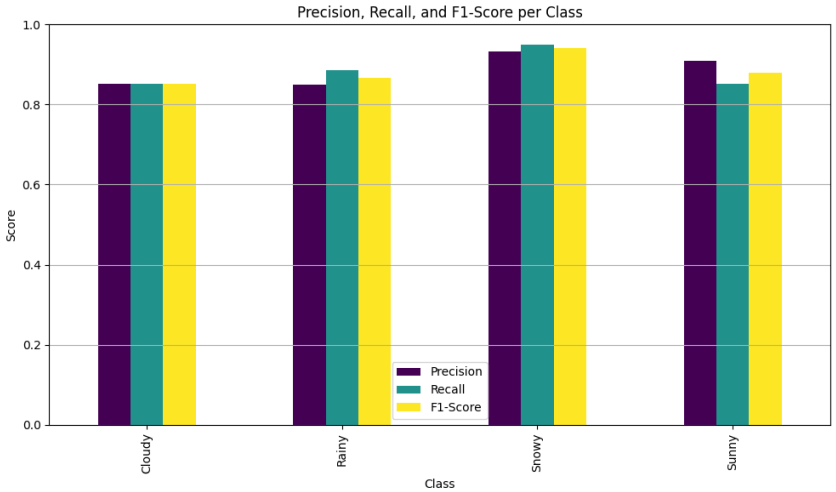
```
precision, recall, f1, _ =
precision_recall_fscore_support(y_test, y_pred)
metrics_df = pd.DataFrame({
    'Class': target_le.classes_,
    'Precision': precision,
    'Recall': recall,
```

repository.ibrahimy.ac.id

Segmen Program 4. 10 (lanjutan)

```
        'F1-Score': f1 })
    metrics_df.set_index('Class').plot(kind='bar',
    figsize=(10, 6), colormap='viridis')
    plt.title('Precision, Recall, and F1-Score per
    Class')
    plt.ylabel('Score')
    plt.ylim(0, 1)
    plt.grid(axis='y')
    plt.tight_layout()
    plt.show()
```

Segmen Program 4.10 untuk mengevaluasi performa model klasifikasi KNN dengan menghitung nilai *precision*, *recall*, dan *F1-score* pada data uji. Proses evaluasi dilakukan menggunakan fungsi *precision_recall_fscore_support* (y_*test*, y_*pred*) yang menghasilkan nilai metrik evaluasi untuk setiap kelas. Hasil perhitungan disajikan dalam bentuk tabel *DataFrame* dengan kolom *Class*, *Precision*, *Recall*, dan *F1-Score*. Selanjutnya, data divisualisasikan dalam grafik batang yang sudah ditampilkan pada gambar 4.8, menggunakan *plot(kind='bar')* dengan *colormap viridis*, dilengkapi judul, label sumbu, batas nilai 0–1, serta garis *grid* sehingga performa tiap kelas dapat diamati dengan lebih jelas dan informatif.



Gambar 4. 8 *Precision, Recall, F1-Score*

repository.ibrahimy.ac.id

1. Presisi *(precision)*

Precision menggambarkan proporsi prediksi positif yang benar dari seluruh prediksi positif yang dihasilkan model. Dalam konteks penelitian ini, jika model memprediksi 10 kali cuaca hujan dan 7 di antaranya sesuai dengan kondisi sebenarnya, maka *precision* dihitung 7/10 = 0,7. Nilai tersebut menunjukkan tingkat ketepatan model dalam memprediksi hujan dibandingkan dengan seluruh prediksi hujan yang dibuat. Untuk menghitung nilai presisi, maka dapat menggunakan rumus 4.2 rumus *precision*, yaitu metrik evaluasi yang mengukur tingkat ketepatan model dalam memprediksi kelas positif. *Precision* menunjukkan proporsi data yang diprediksi positif dan benar-benar positif dibandingkan dengan semua data yang diprediksi positif.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Cloudy\ Precision = \frac{85.3}{85.3 + 14.7} = \frac{85.2}{100} = \mathbf{0.883}\ (85{,}3\%)$$

$$Rainy\ Precision = \frac{88.6}{88.6 + 15.6} = \frac{88.6}{104.4} = \mathbf{0.850}\ (85{,}0\%)$$

$$Snowy\ Precision = \frac{94.9}{94.9 + 7.4} = \frac{94.9}{102.3} = \mathbf{0.927}\ (92.7\%)$$

$$Sunny\ Precision = \frac{85.2}{85.2 + 8.3} = \frac{85.2}{93.5} = \mathbf{0.911}\ (91.1\%)$$

2. Sensitivitas (*Recall*)

Recall merupakan indikator yang menunjukkan kemampuan model dalam mengenali kondisi cuaca positif dari seluruh kejadian positif yang sebenarnya

repository.ibrahimy.ac.id

ada. Sebagaimana yang yang sudah ditampilkan pada rumus 4.3 rumus *Recall*, contohnya, jika terdapat 10 kejadian hujan yang benar-benar terjadi dan model berhasil mendeteksi 7 di antaranya, maka recall bernilai 7/10 atau 0,7. Hal ini berarti model mampu mengidentifikasi 70% dari total kejadian hujan yang sesungguhnya.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

$$Cloudy\ Recall = \frac{85.3}{85.2 + 14.7}\ x\ 100 = \mathbf{85.3}\%$$

$$Rainy\ Recall = \frac{88.6}{88.6 + 11.4}\ x\ 100 = \mathbf{88.6}\%$$

$$Snowy\ Recall = \frac{94.9}{94.9 + 5.1}\ x\ 100 = \mathbf{94.9}\%$$

$$Sunny\ Recall = \frac{85.2}{85.2 + 14.8}\ x\ 100 = \mathbf{85.2}\%$$

3. *F1-Score*

F1-score merupakan nilai yang diperoleh dari rata-rata harmonis antara *precision* dan *recall*. Metrik ini sangat berguna ketika distribusi data antar kelas tidak seimbang, karena mampu memberikan penilaian yang mempertimbangkan keseimbangan antara ketepatan prediksi (*precision*) dan kemampuan model dalam menemukan seluruh kasus positif yang ada (*recall*) sebagaimana yang tertampil pada rumus 4.4 rumus *F1-Score*. Dengan demikian, *F1-score* membantu memberikan gambaran kinerja model secara lebih menyeluruh dibandingkan hanya melihat salah satu metrik saja.

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

$$F1 - Score = 2\ x\ \frac{Precision * Recall}{Precision + Recall} \quad (4.4)$$

$$Cloudy\ F1 - Score = 2\ x\ \frac{0.853\ x\ 0.853}{0.853 + 0.853} = \frac{0.727}{1.706} = \mathbf{0.853}\ (85{,}3\%)$$

$$Rainy\ F1 - Score = 2\ x\ \frac{0.886\ x\ 0.886}{0.886\ + 0.886} = \frac{0.785}{1.772} = \mathbf{0.886}\ (88{,}6\%)$$

$$Snowy\ F1 - Score = 2\ x\ \frac{0.949\ x\ 0.949}{0.949 + 0.949} = \frac{0.900}{1.898} = \mathbf{0.949}\ (94{,}9\%)$$

$$Sunny\ F1 - Score = 2\ x\ \frac{0.852\ x\ 0.852}{0.852 + 0.852} = \frac{0.726}{1.704} = \mathbf{0.852}\ (85{,}2\%)$$

Nilai-nilai tersebut akan disimpan dalam sebuah tabel (*DataFrame*) dan ditampilkan dalam bentuk grafik batang menggunakan *matplotlib.* Grafik ini memudahkan kita melihat seberapa baik model memprediksi tiap kelas, dengan rentang nilai dari 0 hingga 1.

k. *Bar Chart Recall*

Segmen Program 4. 11 *Bar Chart Recall*

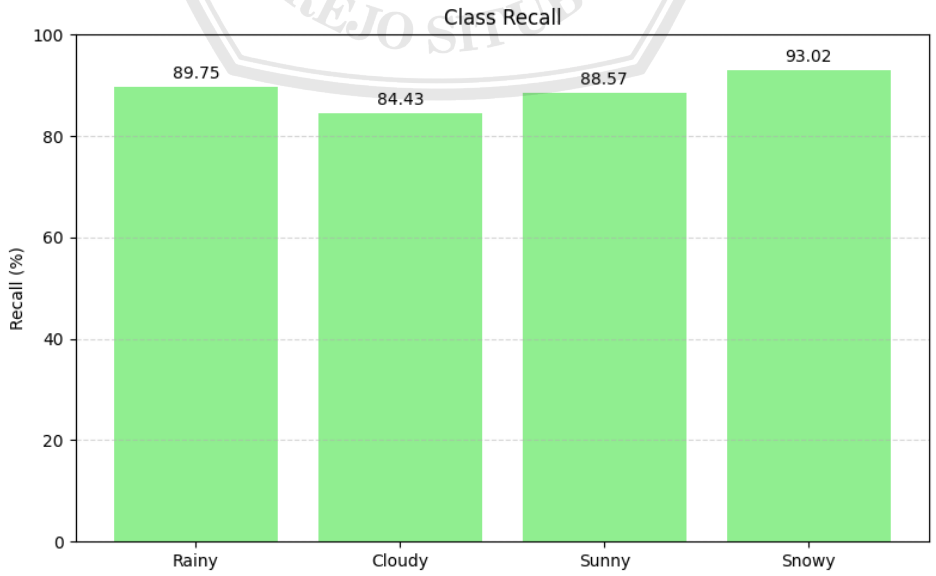
```
import matplotlib.pyplot as plt
# Data recall berdasarkan hasil sebelumnya
labels = ['Rainy', 'Cloudy', 'Sunny', 'Snowy']
recall_scores = [89.75, 84.43, 88.57, 93.02]
# Membuat bar chart
plt.figure(figsize=(8, 5))
bars = plt.bar(labels, recall_scores,
color='lightgreen')
# Tambahkan label angka di atas setiap batang
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2,
yval + 1, f'{yval:.2f}', ha='center', va='bottom',
fontsize=10)
plt.ylim(0, 100)
```

repository.ibrahimy.ac.id

Segmen Program 4. 11 (Lanjutan)

```
        plt.ylabel('Recall (%)')
        plt.title('Class Recall')
        plt.grid(axis='y', linestyle='--', alpha=0.5
        plt.tight_layout()plt.show()
        # Tambahkan label angka di atas setiap batang
        for bar in bars:
            yval = bar.get_height()
            plt.text(bar.get_x() + bar.get_width() / 2, yval +
        1,     f'{yval:.2f}',     ha='center',     va='bottom',
        fontsize=10)
        # Pengaturan tampilan
        plt.ylim(0, 100)
        plt.ylabel('Recall (%)')
        plt.title('Class Recall')
        plt.grid(axis='y', linestyle='--', alpha=0.5)
        plt.tight_layout()
plt.show()
```

Segmen Program 4.11 menampilkan grafik batang nilai recall untuk setiap kelas cuaca, yaitu Rainy, Cloudy, Sunny, dan Snowy. Grafik ditampilkan dengan batang berwarna hijau, disertai label nilai di atas masing-masing batang, serta dilengkapi judul, label sumbu Y, dan garis bantu horizontal yang telah ditampilkan pada gambar 4.9 Grafik *Recall*.



Gambar 4. 9 Grafik *Recall*

repository.ibrahimy.ac.id

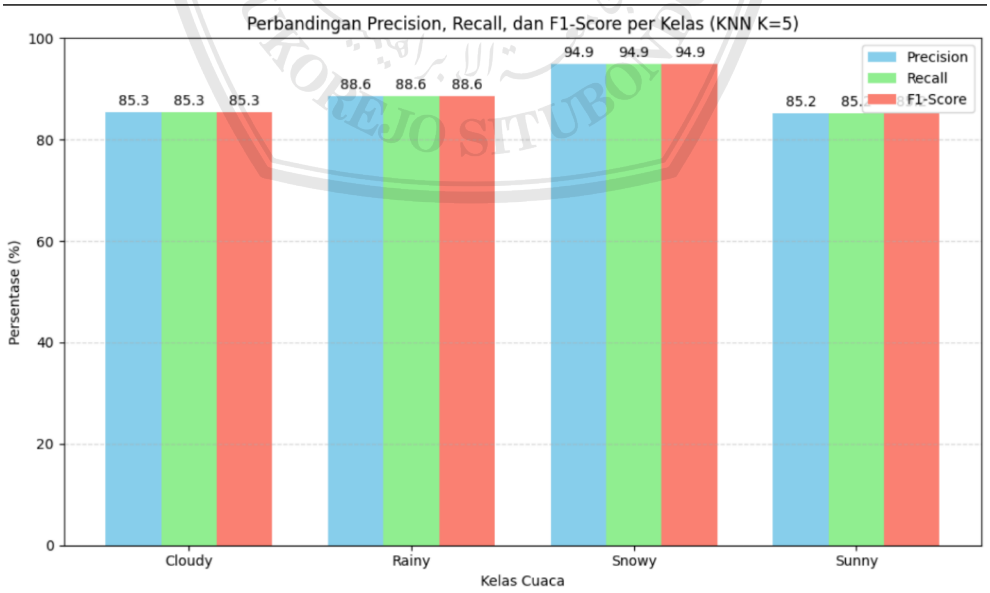
l. Hasil Keseluruhan

Segmen Program 4.12 Grafik Batang Keseluruan

```
import matplotlib.pyplot as plt
import numpy as np

# Data sesuai perhitungan kamu
labels = ['Cloudy', 'Rainy', 'Snowy', 'Sunny']
precision_scores = [85.3, 88.6, 94.9, 85.2]
recall_scores = [85.3, 88.6, 94.9, 85.2]
f1_scores = [85.3, 88.6, 94.9, 85.2]

x = np.arange(len(labels))  # posisi sumbu X
width = 0.25  # lebar batang

fig, ax = plt.subplots(figsize=(10, 6))

# Plot batang
rects1 = ax.bar(x - width, precision_scores, width,
label='Precision', color='skyblue')
rects2 = ax.bar(x, recall_scores, width,
label='Recall', color='lightgreen')
rects3 = ax.bar(x + width, f1_scores, width, label='F1-
Score', color='salmon')

# Label dan judul
ax.set_ylabel('Persentase (%)')
ax.set_xlabel('Kelas Cuaca')
   ax.set_title('Perbandingan Precision, Recall, dan
F1-Score per Kelas (KNN K=5)')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.set_ylim(0, 100)
ax.legend()

# Tambahkan nilai di atas batang
def autolabel(rects):
 for rect in rects:
        height = rect.get_height()
        ax.annotate(f'{height:.1f}%',
                    xy=(rect.get_x()                +
rect.get_width() / 2, height),
                    xytext=(0, 3),
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)
autolabel(rects3)
plt.grid(axis='y', linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```

repository.ibrahimy.ac.id

Segmen Program 4.12 berfungsi membuat grafik batang (*bar chart*) untuk membandingkan *Precision*, *Recall*, dan *F1-Score* pada empat kelas cuaca: *Cloudy*, *Rainy*, *Snowy*, dan *Sunny*. Data dimasukkan ke tiga daftar terpisah, lalu ditampilkan berdampingan denganwarna berbeda. Label persentase ditambahkan di atas setiap batang, sumbu diberi nama, judul ditetapkan, serta *grid horizontal* dibuat agar hasilnya lebih rapi dan mudah dibaca. Grafik ini membandingkan *Precision*, *Recall*, dan *F1-Score* untuk empat kelas cuaca (*Cloudy*, *Rainy*, *Snowy*, *Sunny*) pada model KNN K=5. Setiap kelas memiliki tiga batang yang menunjukkan nilai evaluasi dalam *persentase*, dengan hasil identic untuk ketiga metrik. Seperti yang ditampilkan pada gambar 4.10 *Snowy* memiliki kinerja tertinggi (94,9%), disusul Rainy (88,6%), *Cloudy* (85,3%), dan *Sunny* (85,2%), menunjukkan model paling akurat pada kelas *Snowy* dan sedikit lebih rendah pada *Sunny*.



Gambar 4. 10 Grafik Batang Hasil

repository.ibrahimy.ac.id

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, penerapan algoritma *K-Nearest Neighbors* (K-NN) untuk klasifikasi kondisi cuaca berdasarkan parameter suhu, kelembapan, dan curah hujan menunjukkan kinerja yang cukup optimal. Dengan memanfaatkan dataset berjumlah 13.200 data dari Kaggle dan nilai parameter K=5, model mampu mencapai tingkat akurasi sebesar 88,50%, dengan nilai *precision*, *recall*, dan *F1-score* yang konsisten di berbagai kelas cuaca. Hasil ini membuktikan bahwa metode K-NN efektif dalam mengidentifikasi pola cuaca dan dapat digunakan untuk membantu pengambilan keputusan yang memerlukan prediksi kondisi atmosfer secara cepat dan akurat. Penelitian ini juga membuktikan bahwa penggunaan teknik *instance-based learning* seperti K-NN dapat mengolah data cuaca dalam jumlah besar secara efisien, tanpa memerlukan model matematis yang kompleks. Proses pra-pemrosesan data, pemilihan nilai K yang optimal, serta evaluasi berbasis *confusion matrix* berkontribusi terhadap performa model yang stabil. Dengan demikian, sistem klasifikasi cuaca berbasis K-NN berpotensi diterapkan pada berbagai sektor seperti pertanian, transportasi, dan mitigasi bencana.

5.2 Saran

Penelitian selanjutnya disarankan untuk menguji algoritma K-NN dengan variasi parameter K dan metode pengukuran jarak yang berbeda, seperti *Manhattan distance* atau *Minkowski distance*, untuk melihat pengaruhnya terhadap hasil

repository.ibrahimy.ac.id

klasifikasi. Selain itu, perlu dilakukan perbandingan dengan algoritma *machine learning* lainnya seperti *Decision Tree*, *Random Forest*, atau *Support Vector Machine* guna memperoleh gambaran yang lebih komprehensif terkait performa masing-masing metode. Pengembangan sistem klasifikasi cuaca ini juga dapat diarahkan pada integrasi data cuaca *real-time* melalui API dari badan meteorologi, sehingga prediksi yang dihasilkan lebih aktual. Selain itu, penggunaan teknik optimasi seperti *feature selection* atau *dimensionality reduction* dapat membantu meningkatkan kecepatan pemrosesan tanpa mengurangi akurasi model. Implementasi antarmuka yang ramah pengguna juga akan memudahkan pemanfaatan sistem ini oleh masyarakat umum maupun pihak terkait di lapangan.

Ibrahimy LIBRARY
Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id

Ibrahimy LIBRARY

DAFTAR PUSTAKA

[1] A. N. Kirana, B. Nurhakim, S. E. Permana, W. Prihartono, and G. Dwilestari, "IMPLEMENTASI ALGORITMA NAIVE BAYES UNTUK MEMPREDIKSI CUACA MENGGUNAKAN RAPIDMINER," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 2, pp. 1637–1642, 2024.

[2] A. F. Rodli, S. E. Nikma Yucha, and M. SM, *Manajemen Kinerja Institusi Perguruan Tinggi*. CV Rey Media Grafika, 2022.

[3] E. R. Septiana, F. A. Fiolana, and D. Erwanto, "Klasifikasi Kualitas Citra Kedelai Hitam (Malika) Menggunakan Metode K-Nearest Neighbor," *JEECOM J. Electr. Eng. Comput.*, vol. 4, no. 2, 2022.

[4] A. M. N. Sari, "ANALISA POTENSI KETERSEDIAAN AIR DAN KAPASITAS SIMPAN AIR PADA PERKEBUNANAN KELAPA SAWIT DI LAHAN GAMBUT." Institut Pertanian Stiper Yogyakarta, 2024.

[5] M. L. Laia and Y. Setyawan, "Perbandingan hasil klasifikasi curah hujan menggunakan metode SVM dan NBC," *J. Stat. Ind. dan Komputasi*, vol. 5, no. 02, pp. 51–61, 2020.

[6] E. Erwin *et al.*, *Transformasi Digital*. PT. Sonpedia Publishing Indonesia, 2023.

[7] F. Putra, H. F. Tahiyat, R. M. Ihsan, R. Rahmaddeni, and L. Efrizoni, "Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia: Application of K-Nearest Neighbor Algorithm Using Wrapper as Preprocessing for Determination of Human Weight Information," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 273–281, 2024.

[8] F. Sulianta, *Basic Data Mining from A to Z*. Feri Sulianta, 2023.

[9] M. F. S. Aldy, D. S. Angreni, M. Y. Pusadan, and W. Wirdayanti,

Universitas Ibrahimy
Ibrahimy Repository
repository.ibrahimy.ac.id

"KLASIFKASI CURAH HUJAN MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR (KNN) DI SULAWESI TENGAH," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 4, pp. 2316–2324, 2024.

[10] V. R. Danestiara, "Algoritma k-Nearest Neighbor Classifier untuk Prediksi Curah Hujan di Kabupaten Bandung," *SisInfo*, vol. 5, no. 1, pp. 7–15, 2023.

[11] 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.

[12] E. Priyanto, E. Daniati, and A. Ristyawan, "Implementasi Metode K-Nearest Neighbor Untuk Memprediksi Kondisi Cuaca," in *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 2024, pp. 376–383.

[13] D. Dandy, "Klasifikasi Cuaca dengan Menggunakan Algoritma K-Nearest Neighbor." Universitas Multi Data Palembang, 2023.

[14] N. Nursobah, S. Lailiyah, B. Harpad, and M. Fahmi, "Penerapan Data Mining Untuk Prediksi Perkiraan Hujan dengan Menggunakan Algoritma K-Nearest Neighbor," *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, pp. 1395–1400, 2022.

[15] R. Aprilian, R. Habibi, and M. Y. H. Setyawan, *Algoritma KNN dalam memprediksi cuaca untuk menentukan tanaman yang cocok sesuai musim*. Kreatif, 2020.

[16] Y. Ardilla *et al.*, *Data Mining dan Aplikasinya*. Penerbit Widina, 2021.

[17] I. Werdiningsih, M. Kom, B. Nuqoba, M. Kom, and S. S. Muhammadun, *Data Mining Menggunakan Android, Weka, dan SPSS*. Airlangga University Press, 2020.

[18] S. T. Yahya, *Data Mining*. CV Jejak (Jejak Publisher), 2022.

[19] S. Maesaroh, L. Afiyati, L. Hakim, and Y. S. Sari, "Bahasa Pemrograman

repository.ibrahimy.ac.id

Python," *Sada Kurnia Pustaka*, 2024.

[20] N. S. N. Az-zahrani, H. K. A. Eloi, F. Salim, A.-Z. A. Ramadhani, C. Meysyanti, and L. N. A. Purwantiningsih, *Python untuk Analisis Data*. SIEGA Publisher, 2025.

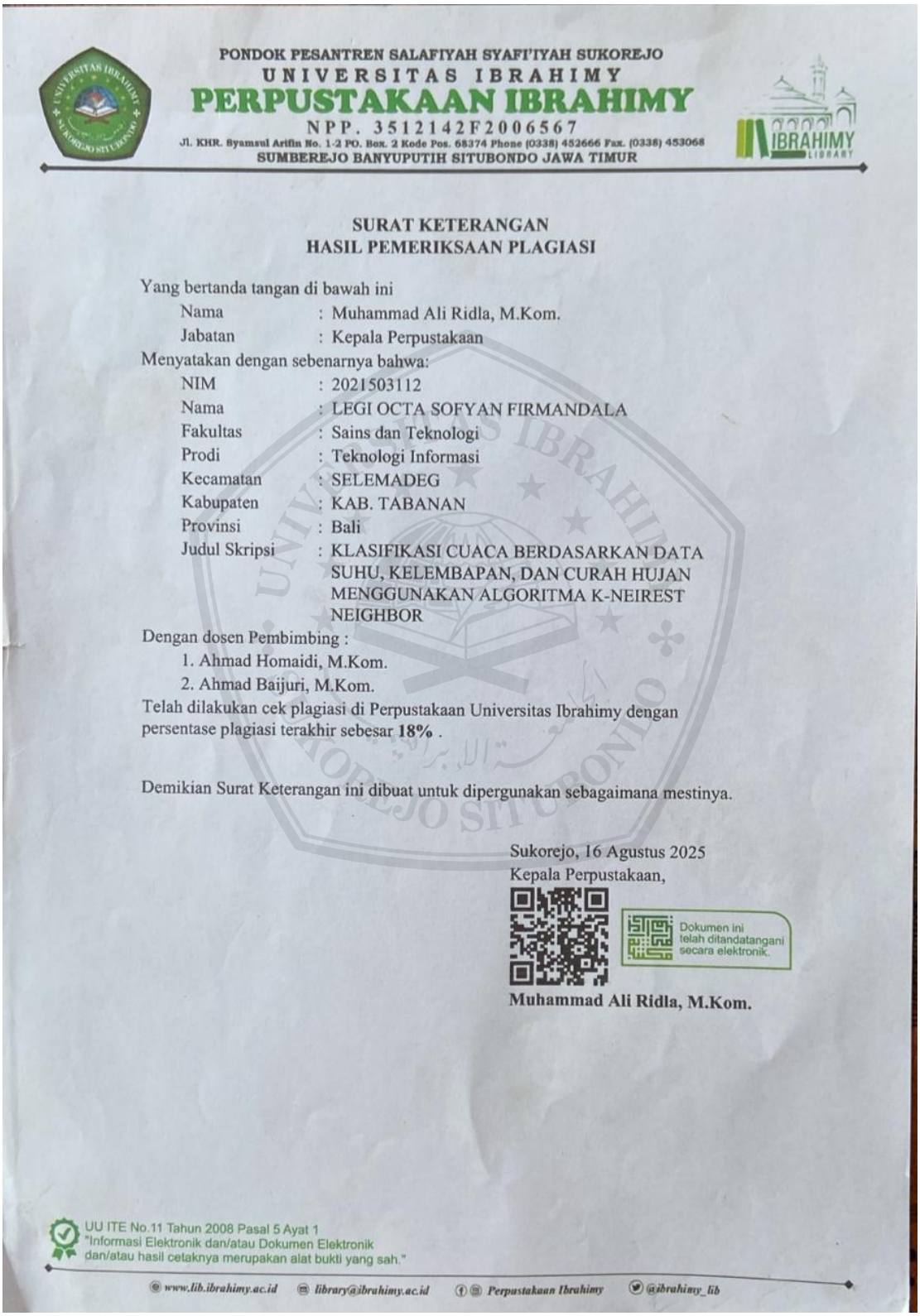
[21] A. Muhlis, "DEEP LEARNING DALAM PENDIDIKAN DAN ARTIFICIAL INTELLEGENCE." Yayasan Putra Adi Dharma, 2025.

[22] S. Junaidi *et al.*, *Buku Ajar Machine Learning*. PT. Sonpedia Publishing Indonesia, 2024.

[23] R. R. Burhanuddin, "KLASIFIKASI penyakit padi melalui citra daun menggunakan metode naive bayes," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024.

repository.ibrahimy.ac.id

LAMPIRAN



PONDOK PESANTREN SALAFIYAH SYAFI'IYAH SUKOREJO
UNIVERSITAS IBRAHIMY
PERPUSTAKAAN IBRAHIMY
NPP. 3512142F2006567
Jl. KHR. Syamsul Arifin No. 1-2 PO. Box. 2 Kode Pos. 68374 Phone (0338) 452666 Fax. (0338) 453068
SUMBEREJO BANYUPUTIH SITUBONDO JAWA TIMUR

SURAT KETERANGAN
HASIL PEMERIKSAAN PLAGIASI

Yang bertanda tangan di bawah ini

Nama	:	Muhammad Ali Ridla, M.Kom.
Jabatan	:	Kepala Perpustakaan

Menyatakan dengan sebenarnya bahwa:

NIM	:	2021503112
Nama	:	LEGI OCTA SOFYAN FIRMANDALA
Fakultas	:	Sains dan Teknologi
Prodi	:	Teknologi Informasi
Kecamatan	:	SELEMADEG
Kabupaten	:	KAB. TABANAN
Provinsi	:	Bali
Judul Skripsi	:	KLASIFIKASI CUACA BERDASARKAN DATA SUHU, KELEMBAPAN, DAN CURAH HUJAN MENGGUNAKAN ALGORITMA K-NEIREST NEIGHBOR

Dengan dosen Pembimbing :
1. Ahmad Homaidi, M.Kom.
2. Ahmad Baijuri, M.Kom.

Telah dilakukan cek plagiasi di Perpustakaan Universitas Ibrahimy dengan persentase plagiasi terakhir sebesar **18%** .

Demikian Surat Keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Sukorejo, 16 Agustus 2025
Kepala Perpustakaan,

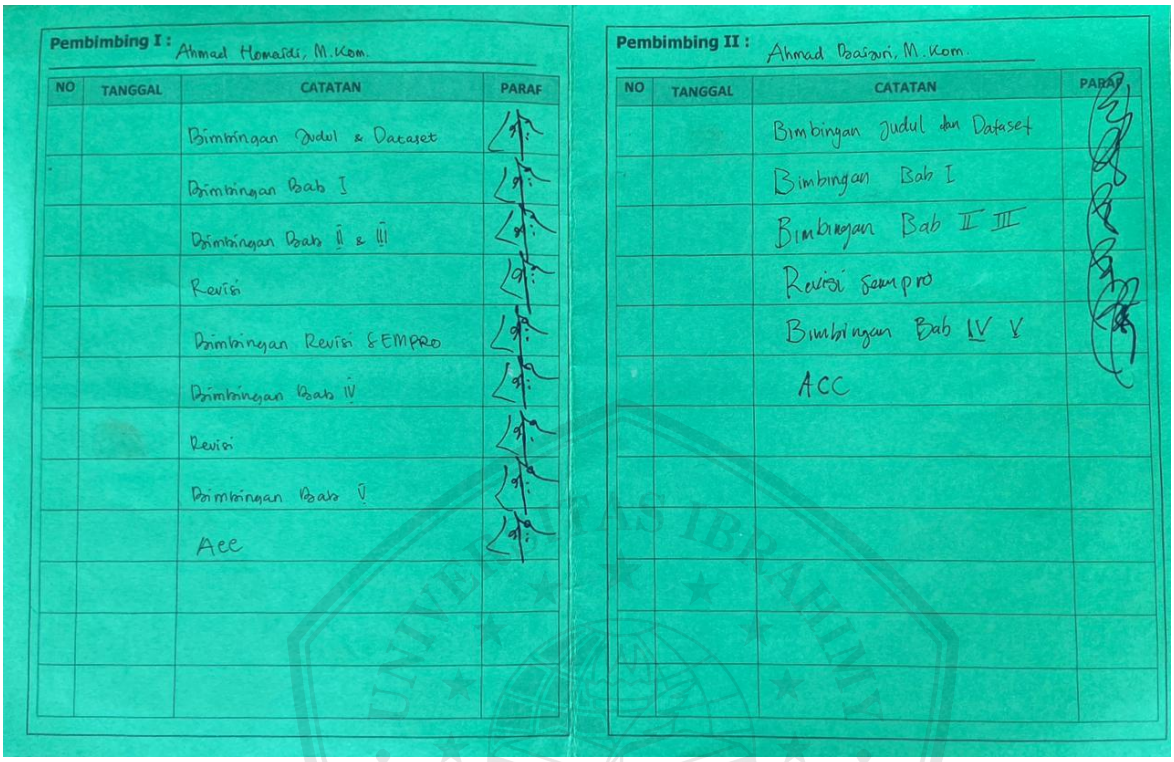
Dokumen ini telah ditandatangani secara elektronik.

Muhammad Ali Ridla, M.Kom.

UU ITE No.11 Tahun 2008 Pasal 5 Ayat 1
"Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti yang sah."

www.lib.ibrahimy.ac.id library@ibrahimy.ac.id Perpustakaan Ibrahimy @ibrahimy_lib

repository.ibrahimy.ac.id



Pembimbing I : Ahmad Homaidi, M.Kom.

NO	TANGGAL	CATATAN	PARAF
		Bimbingan Judul & Dataset	
		Bimbingan Bab I	
		Bimbingan Bab II & III	
		Revisi	
		Bimbingan Revisi & EMPRO	
		Bimbingan Bab IV	
		Revisi	
		Bimbingan Bab V	
		Acc	

Pembimbing II : Ahmad Baizuri, M.Kom.

NO	TANGGAL	CATATAN	PARAF
		Bimbingan Judul dan Dataset	
		Bimbingan Bab I	
		Bimbingan Bab II III	
		Revisi Sempro	
		Bimbingan Bab IV V	
		ACC	

repository.ibrahimy.ac.id

Ibrahimy LIBRARY

CURICULUM VITAE

Identitas Diri

Nama Lengkap	: Legi Octa Sofyan Firmandala
NPM	: 2021503112
Tempat, Tanggal Lahir	: Situbondo, 12 Oktober 2002
Program Studi	: Teknologi Informasi

Nama Orang Tua

Ayah	: Samsuri
Ibu	: Cukmaina

Latar Belakang Pendidikan

SD/MI	: SDN 4 BAJERA
SMP/MTs	: SMPN 1 SELEMADEG
SMA/SMK/MA	: SMAN 1 SELEMADEG
Latar Organisasi	: - BEM Universitas Ibrahimy
	- PMII Universitas Ibrahimy

Alamat Rumah	: Banyuputih - Situbondo
No. Telepon	: 085858280368
E-mail	: legioctasofyan@gmail.com

Ibrahimy Repository
Universitas Ibrahimy
repository.ibrahimy.ac.id