

LAMPIRAN

DAFTAR RIWAYAT HIDUP

NAMA LENGKAP : ANTONIY SANDI WARDANA
NIM : 061940352354
TEMPAT, TANGGAL LAHIR : JAKARTA, 22 JANUARI 2001
ALAMAT : KOMPLEK TNI-AL BLOK A NO 7
LR.JALESVEVA JAYAMAHE RT 26
RW 09 KEL.PLAJU DARAT
KEC.PLAJU KOTA PALEMBANG
TELEPON : 08997119332

RIWAYAT PENDIDIKAN FORMAL

PENDIDIKAN	NAMA SEKOLAH	TAMAT TAHUN
SD	SD NEGERI 87 PALEMBANG	2013
SMP	SMP NEGERI 24 PALEMBANG	2016
SMA	MA NEGERI 1 PALEMBANG	2019

PENGALAMAN PENELITIAN ATAU PROJEK

NO	NAMA PENELITIAN	TAHUN
1.	<i>SYSTEM MOVIE RECOMMENDATION CONTENT BASED FILTERING ARTIFICIAL INTELLIGENCE MASTERY PROGRAM (MSIB BATCH 2)</i>	2022
2	<i>IMPLEMENTASI RANSOMWARE, DDOS, ARP SPOOFING, DAN BRUTEFORCE SEBAGAI TEKNIK CYBER ATTACK BERBAHAYA PADA SISTEM OPERASI WEB SERVER (MSIB BATCH 3)</i>	2022
3	<i>OPTIMASI GAUSSIAN NAÏVE BAYES DENGAN HYPERPARAMETER TUNING DAN UNIVARIATE FEATURE SELECTION DALAM PREDIKSI CUACA (JURNAL AKREDITASI SINTA 3)</i>	2023

**PENGALAMAN PELATIHAN KERJA ATAU PROGRAM KAMPUS DAN
MAGANG**

NO.	KEGIATAN	PENGALAMAN BIDANG	INSTANSI	PERIODE
1	BALAI LATIHAN KERJA PROVINSI	TEKNIK ELEKTRONIKA (AUDIO VIDEO)	BLKPPT PROVINSI SUMSEL	Oktober- November 2021
2	BALAI LATIHAN KERJA KOTA	TEKNIK REFRIGRASI	BLKPPT KOTA PALEMBANG	Juli 2022- Agustus 2022
3	STUDI INDEPENDENT BERSERTIFIKAT (MSIB BATCH 2)	ARTIFICIAL INTELLIGENCE MASTERY PROGRAM	PT. ORBIT FUTURE ACADEMY	21 Feb 2022 - 22 Juli 2022
4	MAGANG KAMPUS MERDEKA (MSIB BATCH 3)	CYBERSECURITY ANALYST	BALITBANG SDM KEMKOMINFO RI	19 Agu 2022 – 2 Des 2022

Semua data yang saya isikan dan tercantum dalam daftar riwayat hidup ini adalah benar dan dapat dipertanggung jawabkan.

Palembang, Agustus 2023

(ANTONIY SANDI WARDANA)



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139 Telp. 0711-353414

Laman: <http://polsri.ac.id>, Pos El : info@polsri.ac.id



KESEPAKATAN BIMBINGAN TUGAS AKHIR (TA)

Kami yang bertanda tangan dibawah ini,

Pihak Pertama

Nama : Antony Sandi Wardana
NIM : 061940352354
Jurusan : Teknik Elektro
Program Studi : Sarjana Terapan Teknik Telekomunikasi

Pihak Kedua

Nama : Lindawati, S.T., M.T.I.
NIP : 197105282006042001
Jurusan : Teknik Elektro
Program Studi : Sarjana Terapan Teknik Telekomunikasi

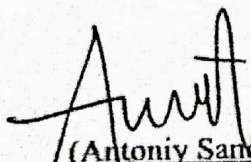
Pada hari ini Selasa tanggal 2 Mei 2023 telah sepakat untuk melakukan konsultasi bimbingan Tugas Akhir (TA).

Konsultasi bimbingan sekurang- kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Senin pukul 10.00 - 16.00 tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir.

Pihak Pertama

Palembang, 2 Mei 2023
Pihak Kedua


(Antony Sandi Wardana)
NIM. 061940352354


(Lindawati, S.T., M.T.I.)
NIP. 197105282006042001

Mengetahui
Ketua Jurusan Teknik Elektro


(Ir. Iskandar Lutfi, M.T.)
NIP. 196501291991031002



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139 Telp. 0711-353414

Laman: <http://polsri.ac.id>, Pos El : info@polsri.ac.id



KESEPAKATAN BIMBINGAN TUGAS AKHIR (TA)

Kami yang bertanda tangan dibawah ini,

Pihak Pertama

Nama : Antony Sandi Wardana
NIM : 061940352354
Jurusan : Teknik Elektro
Program Studi : Sarjana Terapan Teknik Telekomunikasi

Pihak Kedua

Nama : Mohammad Fadhli, S.Pd., M.T.
NIP : 199004032018031001
Jurusan : Teknik Elektro
Program Studi : Sarjana Terapan Teknik Telekomunikasi

Pada hari ini Selasa tanggal 2 Mei 2023 telah sepakat untuk melakukan konsultasi bimbingan Tugas Akhir (TA).

Konsultasi bimbingan sekurang- kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Kerja pukul 10-16..... tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Tugas Akhir.

Pihak Pertama

(Antony Sandi Wardana)
NIM. 061940352354

Palembang, 2 Mei 2023
Pihak Kedua

(Mohammad Fadhli, S.Pd., M.T.)
NIP. 199004032018031001

Mengetahui
Ketua Jurusan Teknik Elektro

(Ir. Iskandar Lutfi, M.T.)
NIP. 196501291991031002



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 Fax. 0711-355918

Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



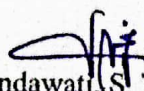
LEMBAR BIMBINGAN LAPORAN AKHIR

Nama : Antony Sandi Wardana
 NIM : 061940352354
 Jurusan/Program Studi : Teknik Elektro / DIV Teknik Telekomunikasi
 Judul Laporan Akhir : Optimalisasi Model *Gaussian Naïve Bayes* Menggunakan Metode *Hyperparameter Tuning* dan *Univariate Feature Selection* Untuk Meningkatkan Akurasi Prediksi Cuaca Kota Palembang
 Pembimbing I : Lindawati, S. T., M. T. I.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	8/4 2023	Konsultasi Dataset	<i>Ld</i>
2.	15/4 2023	Konsultasi Pemrograman ML	<i>Ld</i>
3.	15/5 2023	Konsultasi Model Naive Bayes	<i>Ld</i>
4.	22/5 2023	Konsultasi Optimasi Model	<i>Ld</i>
5.	24/5 2023	Bimbingan Bab I - II	<i>Ld</i>
6.	30/5 2023	Revisi Bab I	<i>Ld</i>
7.	6/6 2023	Konsultasi Hasil Optimasi 90%	<i>Ld</i>
8.	21/6 2023	Konsultasi Hasil Optimasi 98%	<i>Ld</i>
9.	3/7 2023	Bimbingan Bab III	<i>Ld</i>
10.	11/7 2023	Revisi Bab III	<i>Ld</i>
11.	18/7 2023	Bimbingan Bab IV	<i>Ld</i>
12.	2/8 2023	Bimbingan Jurnal	<i>Ld</i>
13.	3/8 2023	Konsultasi Uji Prediksi Data Baru	<i>Ld</i>
14.	4/8 2023	Bimbingan Codingan Akhir	<i>Ld</i>

15.	10/8 2023	Revisi Bab I-IV	SL
16.	11/8 2023	ACC	SL
17.			
18.			
19.			
20.			

Palembang, Agustus 2023
Koordinator Program Studi
Sarjana Terapan Teknik
Telekomunikasi


Lindawati, S. T., M. T. I.
NIP 197105282006042001



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 Fax. 0711-355918

Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



LEMBAR BIMBINGAN LAPORAN AKHIR

Nama : Antony Sandi Wardana
 NIM : 061940352354
 Jurusan/Program Studi : Teknik Elektro / DIV Teknik Telekomunikasi
 Judul Laporan Akhir : Optimalisasi Model *Gaussian Naïve Bayes* Menggunakan Metode *Hyperparameter Tuning* dan *Univariate Feature Selection* Untuk Meningkatkan Akurasi Prediksi Cuaca Kota Palembang
 Pembimbing II : Mohammad Fadhli, S.Pd., M. T.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	15/4 2023	Konsultasi Dataset	
2.	19/4 2023	Konsultasi Codingan ML	
3.	15/5 2023	Konsultasi Optimasi Model	
4.	30/5 2023	Konsultasi Hasil Optimasi 96%	
5.	7/6 2023	Bimbingan Hasil codingan	
6.	12/6 2023	konsultasi Hasil optimasi 98%	
7.	27/6 2023	Bimbingan Bab I	
8.	9/7 2023	Revisi Bab I	
9.	16/7 2023	Konsultasi Hasil Prediksi	
10.	23/7 2023	Konsultasi Hasil codingan	
11.	30/7 2023	Bimbingan Uji Prediksi Baru	
12.	2/8 2023	Bimbingan Jurnal	
13.	9/8 2023	Bimbingan Bab I-V	
14.	10/8 2023	Revisi Bab I, III, IV	

15.	11/8 2023	Acc	
16.			
17.			
18.			
19.			
20.			

Palembang, Agustus 2023
Koordinator Program Studi
Sarjana Terapan Teknik
Telekomunikasi


Lindawati S. T., M. T. I.
NIP 197105282006042001



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

**REKOMENDASI UJIAN TUGAS AKHIR**

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

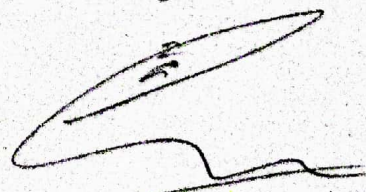
Nama	: Antoniy Sandi Wardana
NIM	: 061940352354
Jurusan/Program Studi	: Teknik Elektro/DIV Teknik Telekomunikasi
Judul Laporan Kerja Akhir	: Optimalisasi Model <i>Gaussian Naïve Bayes</i> Menggunakan Metode <i>Hyperparameter Tuning</i> Dan <i>Univariate Feature Selection</i> Untuk Meningkatkan Akurasi Prediksi Cuaca Kota Palembang

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Tugas Akhir pada Tahun Akademik 2022/2023

Pembimbing I,


(Lindawati, S.N., M.T.)
NIP 197105282006042001

Palembang, Agustus 2023
Pembimbing II,


(Mohammad Fadhli, S.Pd., M.T.)
NIP 199004032018031001



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA**

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 Fax. 0711-355918

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



PELAKSANAAN REVISI UJIAN TUGAS AKHIR

Mahasiswa berikut,

Nama : Antoni Sandy Wardana
NIM : 061940352354
Jurusan/Program Studi : Teknik Elektro/DIV Teknik Telekomunikasi
Judul Laporan KP : Optimalisasi Model *Gaussian Naïve Bayes* Menggunakan Metode *Hyperparameter Tuning* Dan *Univariate Feature Selection* Untuk Meningkatkan Akurasi Prediksi Cuaca Kota Palembang

Telah melaksanakan revisi terhadap Ujian Tugas Akhir (TA) yang dilaksanakan pada hari Rabu tanggal 16 Agustus 2023 Pelaksanaan revisi terhadap Ujian Tugas Akhir (TA) tersebut telah disetujui oleh Dosen Penilai yang memberikan revisi:

No.	Komentar	Nama Dosen Penilai	Tanggal	Tanda Tangan
1.	<i>Substansi materi</i>	Ciksadan, S.T., M.Kom NIP. 196809071993031003	<i>08/08/23</i>	<i>[Signature]</i>
2.	<i>Acc</i>	Lindawati, S. T., M.T.I NIP. 197105282006042001	<i>6/8-23</i>	<i>[Signature]</i>
3.	<i>Acc</i>	Eka Susanti, S.T., M.Kom NIP. 197812172000122001	<i>6/8/2023</i>	<i>[Signature]</i>
4.	<i>Acc</i>	Martinus Mujur Rose, S.T., M.T NIP. 197412022008121002	<i>30/8-2023</i>	<i>[Signature]</i>

Palembang, Agustus 2023

Ketua Penilai,

Ciksadan, S.T., M.Kom
NIP. 196809071993031003



Edumatic: Jurnal Pendidikan Informatika

Rektorat Universitas Hamzanwadi Lt. 2
Jalan TGKH Zainuddin Abdul Majid No. 132 Pancor-Selong Kab. Lombok Timur NTB
email: edumatic.pi@gmail.com; Hp. 0877-6182-2150
Website: <http://e-journal.hamzanwadi.ac.id/index.php/edumatic>



SURAT KETERANGAN

No. 025/Edumatic/UH/VIII/2023

Ketua Penyunting Edumatic: Jurnal Pendidikan Informatika menerangkan bahwa:

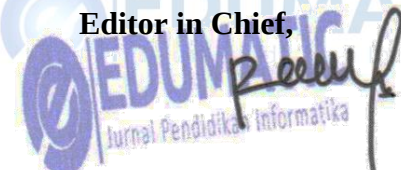
Nama : Lindawati, Mohammad Fadhli, Antoniy Sandi Wardana
Instansi : Program Studi Teknik Telekomunikasi, Politeknik Negeri Sriwijaya, Indonesia
Judul : Optimasi Gaussian Naïve Bayes dengan Hyperparameter Tuning dan Univariate Feature Selection dalam Prediksi Cuaca

Telah melalui proses penyuntingan dan **DITERIMA** untuk dipublikasi. *Manuscript* tersebut akan dipublikasikan pada bulan Desember 2023 di Jurnal **Edumatic: Jurnal Pendidikan Informatika** dengan e-ISSN 2549-7472 **Volume 7 Nomor 2 Desember 2023** yang diterbitkan oleh Universitas Hamzanwadi. **Edumatic: Jurnal Pendidikan Informatika** sudah terakreditasi peringkat 3 (SINTA 3) berdasarkan **SK No. 105/E/KPT/2022** mulai dari Volume 5 Nomor 2 Tahun 2022 sampai Volume 10 Nomor 1 Tahun 2026

Demikian surat keterangan ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Lombok Timur, 5 Agustus 2023

Editor in Chief,


Muhammad Zamroni Uska, M.Kom.

Optimasi *Gaussian Naïve Bayes* Dengan *Hyperparameter Tuning* dan *Univariate Feature Selection* Dalam Prediksi Cuaca

Lindawati¹, Mohammad Fadhli¹, Antony Sandi Wardana^{1,*}

¹ Program Studi Sarjana Terapan Teknik Telekomunikasi, Politeknik Negeri Sriwijaya, Indonesia

* Correspondence: antonisandi400@gmail.com

Copyright: © 2023 by the authors

Received: 02 Agustus 2023 | Revised: 04 Agustus 2023 | Accepted: 06 Agustus 2023 | Published: 02 Desember 2023

Abstrak

Pentingnya melakukan penelitian prediksi cuaca karena perubahan cuaca memiliki pengaruh signifikan dalam kehidupan sehari-hari. Tujuan penelitian ini adalah menerapkan metode klasifikasi *machine learning* yang optimal untuk memprediksi cuaca. Metode yang digunakan adalah model *Gaussian Naïve Bayes* yang telah dioptimasi menggunakan teknik optimasi *Univariate Feature Selection* ANOVA-*f* test dan *Hyperparameter Tuning GridsearchCV*. Data yang digunakan terdiri dari 6454 data cuaca harian di Kota Palembang. Terdapat 5 uji data model *Gaussian Naïve Bayes* sebelum dan setelah dilakukan optimasi. Hasil penelitian menunjukkan bahwa optimasi model berhasil meningkatkan performa dalam memprediksi cuaca. Hasil akurasi tertinggi setelah optimasi mencapai 98.33% pada 644 data uji, meningkat dari akurasi sebelum optimasi yang hanya 96.95%. Sebelum dilakukan optimasi, hasil prediksi untuk kondisi cuaca cerah berawan, mendung/hujan, hujan sedang, dan hujan lebat sesuai dengan data aktual. Namun, terdapat 20 kesalahan prediksi saat dihadapkan pada data yang seharusnya merupakan kondisi hujan sangat lebat. Setelah dilakukan optimasi, jumlah kesalahan prediksi pada data hujan sangat lebat berhasil berkurang menjadi 7 kesalahan. Pendekatan optimasi yang digunakan dalam penelitian ini membantu mencari kombinasi parameter yang paling tepat, dan fitur-fitur yang tidak relevan dapat dieliminasi, sehingga model hanya mempertimbangkan fitur-fitur yang memiliki kontribusi signifikan dalam memprediksi cuaca.

Kata kunci: *prediksi cuaca; gaussian naïve bayes; univariate feature selection; hyperparameter tuning*

Abstract

The importance of conducting weather prediction research is due to the significant influence of weather changes on daily life. The purpose of this study is to apply an optimal machine learning classification method for weather prediction. The method used is the *Gaussian Naïve Bayes* model, which has been optimized using *Univariate Feature Selection* ANOVA-*f* test and *Hyperparameter Tuning GridsearchCV* techniques. The data used consists of 6454 daily weather data in Palembang City. There are 5 tests on the *Gaussian Naïve Bayes* model before and after optimization. The research results show that the optimization of the model successfully improves the performance in weather prediction. The highest accuracy result after optimization reaches 98.33% with 644 test data, an improvement from the pre-optimization accuracy of only 96.95%. Before optimization, the predictions for weather conditions such as sunny, cloudy/rainy, light rain, and heavy rain match the actual data. However, there were 20 prediction errors when dealing with data that should represent very heavy rain conditions. After optimization, the number of prediction errors for the very heavy rain data reduced to seven. The optimization approach used in this research helps find the most suitable parameter combinations and eliminates irrelevant features, allowing the model to consider only significant features in weather prediction.

Keywords: *weather prediction; gaussian naïve bayes; univariate feature selection; hyperparameter tuning*



PENDAHULUAN

Cuaca adalah keadaan fisik udara dan peristiwa perubahan atmosfer yang terjadi dari waktu ke waktu di suatu tempat dan waktu tertentu. Berbagai faktor, seperti suhu rata-rata, kelembaban, kecepatan angin, arah angin, lama penyinaran matahari, curah hujan, dan lain-lain, dapat mempengaruhi perubahan cuaca (Sunarmi et al., 2022). Informasi cuaca yang akurat memungkinkan masyarakat dan berbagai sektor untuk mengambil langkah yang tepat dan adaptasi yang diperlukan untuk menghadapi perubahan cuaca yang mungkin terjadi, sehingga prediksi cuaca menarik untuk diteliti karena perubahan cuaca memiliki peran vital dalam berbagai sektor, seperti industri, sarana penerbangan, pertanian serta kegiatan sehari-hari seperti perencanaan perjalanan, olahraga dan sebagainya (Intan et al., 2021).

Salah satu cara yang dapat digunakan untuk melakukan prediksi cuaca adalah dengan menggunakan kecerdasan buatan. Tujuan utama kecerdasan buatan adalah untuk memahami dan memodelkan proses berpikir manusia, serta untuk membuat algoritma yang dapat digunakan untuk memodelkan proses berpikir manusia yang menjadi salah satu bagian dari *artificial intelligence* yaitu *machine learning* (Susatyono, 2021). Namun terdapat tantangan dalam meningkatkan akurasi prediksi cuaca menggunakan machine learning. Beberapa masalah yang dihadapi adalah data yang sedikit, kompleksitas dan variasi data cuaca harian, termasuk suhu, kelembaban udara, kecepatan angin, dan kondisi cuaca yang beragam (Utami et al., 2021). Selain itu, ada kemungkinan adanya fitur-fitur cuaca yang tidak relevan atau memiliki tingkat korelasi yang rendah dengan kelas cuaca yang diinginkan, sehingga dapat mempengaruhi akurasi prediksi (Yani et al., 2022).

Untuk mengatasi permasalahan tersebut, penelitian ini menggunakan dua pendekatan metode dalam machine learning yaitu *Hyperparameter Tuning GridsearchCV* dan *Univariate Feature Selection ANOVA-f test* dalam optimasi model *Gaussian Naïve Bayes*. *Hyperparameter tuning* akan membantu mencari kombinasi parameter yang paling tepat dalam model *Gaussian Naïve Bayes* untuk meningkatkan kemampuan dalam memprediksi cuaca dengan lebih akurat. Sementara itu, *Feature Selection ANOVA-f test* akan membantu mengidentifikasi fitur-fitur cuaca yang memiliki korelasi tinggi dengan kelas cuaca yang diinginkan, sehingga dapat mengurangi dimensi fitur yang tidak relevan dan meningkatkan efisiensi prediksi (Harpale & Bairagi, 2021; Jebadurai et al., 2022). Dengan melakukan pendekatan tersebut, penelitian yang mereka lakukan memberikan kontribusi dalam pengembangan teknik prediksi cuaca yang lebih efisien dan efektif dalam mengatasi kompleksitas dan variasi data cuaca harian, sehingga informasi cuaca yang lebih handal dapat digunakan untuk perencanaan dan pengambilan keputusan yang lebih baik.

Hasil temuan sebelumnya menunjukkan bahwa beberapa algoritma *machine learning* telah digunakan untuk prediksi cuaca, namun performanya masih belum optimal. Beberapa algoritma mencapai akurasi yang rendah, seperti *Naïve Bayes* dan *C4.5* dengan metode *Ensemble Learning* yang hanya mencapai akurasi sekitar 49,45% dan 41,24% (Yani et al., 2022). Meskipun terdapat algoritma dengan akurasi lebih tinggi, seperti *Deep Learning* dengan akurasi 82.92%, *Generalized Linear Model* dengan akurasi 84.06% (Siregar, 2020). Dan penerapan algoritma *K-Nearest Neighbor* (KNN) dengan akurasi 89% (Rangkuti et al., 2021).. masih ada ruang untuk peningkatan akurasi prediksi cuaca. Semakin tinggi tingkat akurasi dari suatu algoritme, semakin baik kualitas prediksinya. Akurasi menggambarkan seberapa dekat hasil prediksi dengan nilai sebenarnya. Jika akurasi semakin tinggi, artinya algoritma dapat melakukan prediksi dengan tingkat ketepatan yang lebih tinggi, dan hasil prediksinya lebih mendekati kenyataan (Siregar, 2020). Selain masalah akurasi, penelitian sebelumnya juga belum sepenuhnya mencakup analisis tentang efisiensi dan relevansi fitur dalam prediksi cuaca. Dataset yang kurang, fitur-fitur cuaca yang tidak relevan atau memiliki tingkat korelasi rendah dengan kelas cuaca yang diinginkan dapat mempengaruhi akurasi prediksi (Yani et al., 2022; Utami et al., 2021). Oleh karena itu, diperlukan pendekatan yang dapat mengidentifikasi

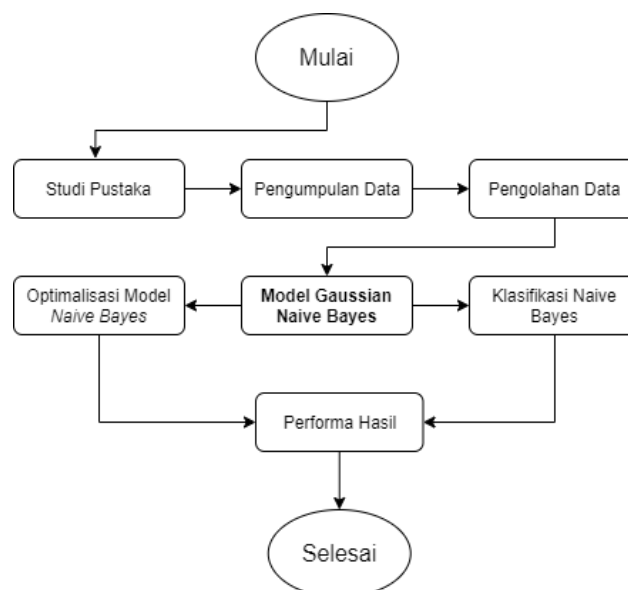
fitur-fitur penting dan relevan sehingga dapat mengurangi dimensi fitur yang tidak relevan dan meningkatkan efisiensi prediksi seperti yang disarankan pada penelitian (Oshodi, 2022). Hasil temuan sebelumnya juga menunjukkan bahwa prediksi cuaca menggunakan *Gaussian Naive Bayes* banyak digunakan (Siregar et al., 2020 ; Yani et al., 2022).dan memiliki kinerja terbaik dengan prediksi akurasi sebesar 84.153% pada penelitian (Oshodi, 2022). Namun, metode *Hyperparameter Tuning* dan *Feature Selection ANOVA-f test* dalam model *Gaussian Naive Bayes* belum banyak digunakan dalam penelitian sebelumnya, dengan melakukan pendekatan ini, diharapkan dapat mengatasi kelemahan dan kekurangan dari penelitian sebelumnya dan menghasilkan prediksi cuaca yang lebih akurat dan efisien.

Penelitian yang dilakukan ini bertujuan untuk mengatasi kompleksitas dan variasi data cuaca yang beragam dengan menggunakan dua pendekatan metode dalam *machine learning*, yaitu *Hyperparameter Tuning* dan *Feature Selection ANOVA-f test* dalam model *Gaussian Naive Bayes*. Selain itu, dengan diterapkan metode ini untuk meningkatkan akurasi prediksi cuaca sehingga informasi cuaca yang lebih handal dapat digunakan untuk perencanaan dan pengambilan keputusan yang lebih baik dalam berbagai sektor dan aktivitas sehari-hari, serta diharapkan hasil prediksi cuaca menjadi lebih akurat, sehingga pada penelitian yang dilakukan ini akan menunjukkan seberapa besar pengaruh optimasi dengan metode yang diterapkan di atas terhadap model *Gaussian Naive Bayes* menggunakan dataset cuaca harian Kota Palembang dalam meningkatkan performa akurasi prediksi cuaca.

METODE

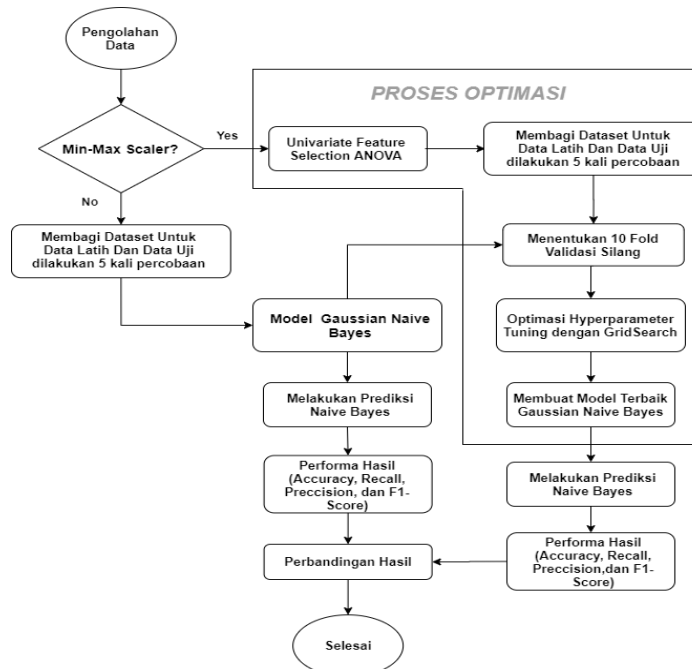
Penelitian ini, dilakukan pengujian untuk meningkatkan akurasi prediksi cuaca di Kota Palembang menggunakan model *Gaussian Naive Bayes*. *Gaussian Naive Bayes* akan diimplementasikan menggunakan bahasa pemrograman *Python*. Pemilihan bahasa pemrograman ini penting karena akan memastikan program yang dikembangkan berfungsi secara optimal (Sari et al., 2020).

Model *Gaussian Naive Bayes* akan dioptimasi dengan menggunakan metode *Univariate Features Selection ANOVA-f test* dan *Hyperparameter GridsearchCV*. Selanjutnya, dilakukan perbandingan performa model sebelum dan setelah dilakukan optimasi untuk melihat seberapa besar peningkatan akurasi yang berhasil dicapai. Alur penelitian bisa dilihat pada gambar 1.



Gambar 1. Alur Penelitian

Studi pustaka adalah proses mempelajari dan mengumpulkan informasi dari berbagai sumber tentang topik penelitian untuk mendapatkan pemahaman yang komprehensif, yang membantu dalam perencanaan dan pelaksanaan penelitian. (Adlini et al., 2022). Pengumpulan data untuk dijadikan dataset penelitian menggunakan data cuaca harian kota Palembang dalam rentang waktu Januari 2000 - April 2023 berjumlah 8400 data pada Stasiun Cuaca klimatologi Palembang yang bersumber dari website <https://dataonline.bmkg.go.id>.



Gambar 2. Proses pengujian model *gaussian naïve bayes*

Pada gambar 2 Pengolahan data (*Pre-processing*) dilakukan dengan memilih variabel independen dan dependen, menghapus data yang tidak diperlukan, membuat label pada data kategorikal, dan menghapus data yang kosong. Setelah dilakukan pre-processing data, didapatkan dataset yang terdiri dari 9 variabel independen (Suhu Min, Suhu Max, Suhu Rata, Kel_Rata, LPM, Kec_Angin Max, Arah Angin, Kec_Angin Rata, dan Curah Hujan) dan 1 variabel dependen (Kondisi Cuaca). Kemudian dataset tersebut dijadikan sebagai data latih dan data uji dengan rasio 90:10, 80:20, 70:30, 60:40, 50:50. Data latih digunakan untuk mendapatkan nilai pada tabel probabilitas dan data uji digunakan untuk menguji tabel probabilitas yang dihasilkan oleh data latih (Susanti et al., 2022). Langkah berikutnya adalah membuat model *Gaussian Naïve Bayes* dan mengoptimasi menggunakan metode *Univariate Features Selection ANOVA-f test*. Metode ini memilih fitur terbaik berdasarkan *test statistic univariate*, di mana *ANOVA-f test* digunakan untuk menghitung skor fitur berdasarkan jarak antara kelas dan dalam kelas (Shakeela et al., 2021). Kemudian, *ANOVA-f test* menghitung hubungan linier antara fitur dan target, dan fitur dengan korelasi tertinggi dipilih (Jebadurai et al., 2022). Setelah pemilihan fitur, dilakukan optimasi menggunakan *Hyperparameter Tuning GridsearchCV* untuk mencari model terbaik. Proses *Cross-Validation* dengan *10 fold* digunakan, di mana model dilatih dan dievaluasi pada kombinasi subset yang berbeda-beda. Keuntungan dari metode ini adalah setiap subset menjadi data uji tepat satu kali dan sebagai data pelatihan sebanyak 9 kali (Aguni et al., 2021), (Alhakeem et al., 2022). Setelah peforma hasil didapatkan, hasil dari sebelum dan sesudah dioptimasi akan di bandingkan berdasarkan tingkat *Accuracy, Precision, Recall, dan F1-Score*.

HASIL DAN PEMBAHASAN

Hasil

Pada penelitian ini, dilakukan optimasi algoritma *Naïve Bayes* menggunakan metode *Univariate Feature Selection ANOVA-f test* dan *Hyperparameter Tuning GridSearchCV* untuk meningkatkan akurasi prediksi cuaca di Kota Palembang. Penggunaan data set cuaca harian kota Palembang awalnya berjumlah 8400, namun setelah dilakukan pembersihan data, seperti menghapus data kosong (*NaN*), menghapus data duplikat, dan menghapus data yang tidak diperlukan pada data kategorikal dalam fitur (Tanggal dan AAT), data set berkurang menjadi 6545 data dengan 9 fitur dan 1 target untuk uji percobaan optimasi model *Naïve Bayes*, yang ditunjukkan pada gambar 3.

<pre><class 'pandas.core.frame.DataFrame'> RangeIndex: 8399 entries, 0 to 8398 Data columns (total 12 columns): # Column Non-Null Count Dtype --- --- 0 Tanggal 8399 non-null object 1 Suhu Min 8258 non-null float64 2 Suhu Max 8299 non-null float64 3 Suhu Rata 8310 non-null float64 4 Kel_Rata 8271 non-null float64 5 LPM 8255 non-null float64 6 Kec_Angin Max 8351 non-null float64 7 Arah Angin 8285 non-null float64 8 Kec_Angin Rata 8353 non-null float64 9 AAT 8330 non-null object 10 Curah Hujan 6828 non-null float64 11 Kondisi Cuaca 8399 non-null object dtypes: float64(9), object(3) memory usage: 787.5+ KB</pre>				<pre><class 'pandas.core.frame.DataFrame'> Int64Index: 6545 entries, 0 to 8396 Data columns (total 10 columns): # Column Non-Null Count Dtype --- --- 0 Suhu Min 6545 non-null float64 1 Suhu Max 6545 non-null float64 2 Suhu Rata 6545 non-null float64 3 Kel_Rata 6545 non-null float64 4 LPM 6545 non-null float64 5 Kec_Angin Max 6545 non-null float64 6 Arah Angin 6545 non-null float64 7 Kec_Angin Rata 6545 non-null float64 8 Curah Hujan 6545 non-null float64 9 Kondisi Cuaca 6545 non-null object dtypes: float64(9), object(1) memory usage: 562.5+ KB None</pre>			
---	--	--	--	--	--	--	--

Gambar 3. Data mentah dan data setelah dilakukan pembersihan untuk pengolahan data

Pada proses pemilihan fitur menggunakan metode *Univariate Feature Selection ANOVA-f test* dari 9 fitur yang dipilih, terdapat 3 fitur dengan bobot rendah, yaitu Kel_Rata, Kec_Angin Max, dan Arah Angin seperti yang ditunjukkan pada gambar 4, lalu dipilih 6 fitur terbaik yang memiliki relevansi tinggi dengan target prediksi kemungkinan kondisi cuaca (Cerah Berawan, Mendung atau Hujan Ringan, Hujan Sedang, Hujan Lebat, dan Hujan Sangat Lebat) sehingga dapat mempengaruhi performa model *Gaussian Naïve Bayes*. Selanjutnya, dilakukan *Hyperparameter Tuning* untuk mengetahui parameter yang optimal untuk model *Gaussian Naïve Bayes*. Proses *Cross-Validation* menggunakan *GridSearchCV* digunakan untuk mengoptimalkan akurasi (Maisat & Ashafidz F.D, 2023). Hasil performa dari percobaan tersebut terlihat pada tabel 1 dan tabel 2, di mana data set dibagi menjadi data uji dan data latih sebanyak 5 kali.

	Feature	Score
0	Suhu Min	175.415480
1	Suhu Max	172.658541
2	Suhu Rata	277.677236
3	Kel_Rata	14.411820
4	LPM	120.346672
5	Kec_Angin Max	14.152377
6	Arah Angin	50.707172
7	Kec_Angin Rata	65.305416
8	Curah Hujan	15219.315846

Fitur Terbaik yang dipilih sebagai optimasi:
 Index(['Suhu Min', 'Suhu Max', 'Suhu Rata', 'LPM', 'Kec_Angin Rata',
 'Curah Hujan'],
 dtype='object')

Gambar 4. Hasil pemilihan fitur terbaik

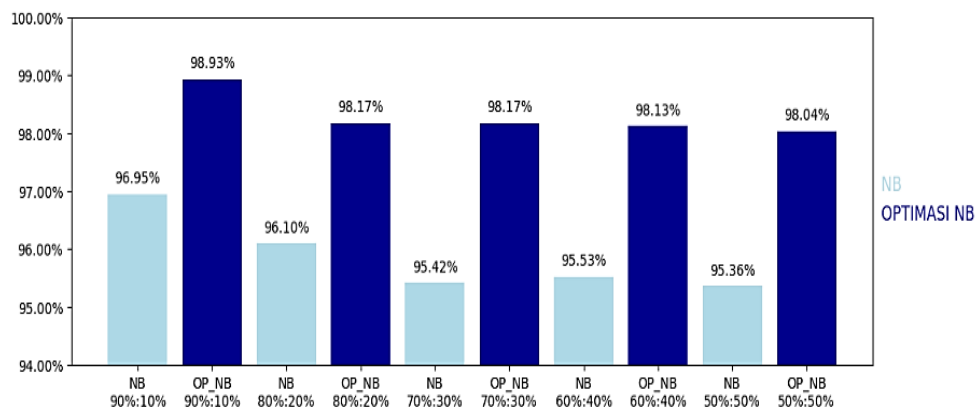
Tabel 1. Rekapitulasi Hasil Percobaan Model Naïve Bayes Sebelum Dioptimasi

Percobaan	Accuracy	Precision	Recall	F1-Score
90%:10%	96,95%	95,56%	98,63%	96,79%
80%:20%	96,10%	94,05%	98,13%	95,63%
70%:30%	95,42%	87,98%	91,96%	89,37%
60%:40%	95,53%	87,67%	92,89%	89,79%
50%:50%	95,36%	86,55%	92,53%	88,85%

Tabel 2. Rekapitulasi hasil Percobaan Model Naïve Bayes Sesudah Di Optimasi

Percobaan	Accuracy	Precision	Recall	F1-Score
90%:10%	98,93%	98,18%	99,52%	98,80%
80%:20%	98,70%	97,38%	99,30%	98,28%
70%:30%	98,17%	93,32%	93,50%	93,25%
60%:40%	98,13%	92,21%	94,36%	93,21%
50%:50%	98,04%	92,10%	93,96%	92,97%

Hasil rekapitulasi nilai pada tabel 1 dan tabel 2 menunjukkan percobaan menggunakan model *Gaussian Naïve Bayes* sebelum dioptimasi, dan sesudah dioptimasi yang dilakukan sebanyak 5 kali pada data set. Dalam percobaan tersebut, nilai akurasi tertinggi diperoleh ketika data uji digunakan sebesar 10% dan data latih sebesar 90%, dengan akurasi mencapai 96.95% sebelum dioptimasi . dan mendapatkan peningkatan menjadi 98.93% setelah dioptimasi Selain itu, nilai *Precision*, *Recall*, dan *F1-Score* tertinggi juga didapatkan pada kondisi yang sama yaitu dengan data uji sebesar 10% dan data latih sebesar 90%, nilai-nilai tersebut tertera pada tabel 1, dan tabel 2. Langkah selanjutnya adalah melakukan perbandingan nilai akurasi dari semua percobaan. Hasil perbandingan menunjukkan bahwa nilai akurasi tertinggi, yaitu 98.93%, diperoleh pada model *Gaussian Naïve Bayes* dioptimasi dengan *Univariate Feature Selection ANOVA-f test* dan *Hyperparameter Tuning GridsearchCV* pada rasio perbandingan dengan 10% data uji dan 90% data latih. Grafik perbandingan nilai akurasi dapat dilihat pada gambar 5.



Gambar 5. Perbandingan seluruh hasil peforma *accuracy* sebelum dioptimasi dan sesudah dioptimasi

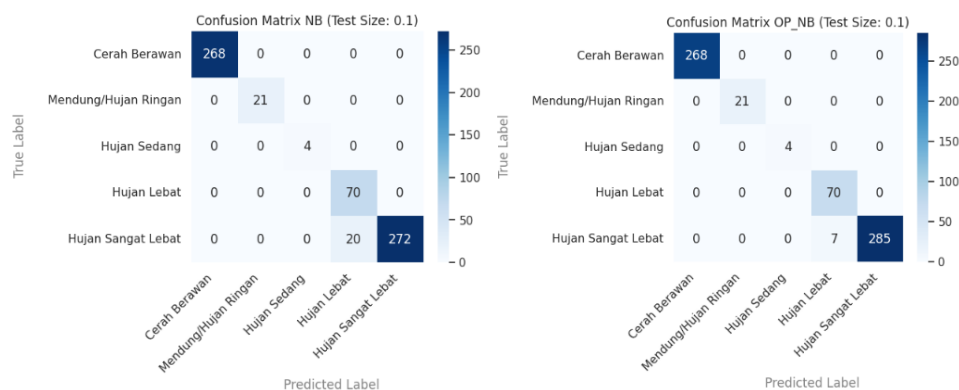
Perbandingan selisih dan peningkatan nilai akurasi yang didapatkan menggunakan model *Gaussian Naïve Bayes* sebelum dioptimasi dan model *Gaussian Naïve Bayes* dioptimasi dengan *Univariate Feature ANOVA* dan *Hyperparameter Tuning GridSearchCV* ditunjukkan pada tabel 3. Selisih peningkatan akurasi tertinggi terjadi pada percobaan menggunakan data latih 70% dan data uji 30% dengan selisih peningkatan sebesar 2.75%.

Tabel 3. Rekapitulasi Hasil Akurasi Seluruh Percobaan

Percobaan	NB	OP_NB	Selisih Peningkatan
90%:10%	96.95%	98.93%	1.98%
80%:20%	96.10%	98.70%	2.6%
70%:30%	95.42%	98.17%	2.75%
60%:40%	95.53%	98.13%	2.6%
50%:50%	95.36%	98.04%	2.68%

Hasil pengujian optimasi yang dilakukan, akurasi tertinggi yang didapatkan pada 10% data uji dan 90% data latih dengan akurasi setelah dioptimasi sebesar 98.93%, dari sebelumnya sebelum dioptimasi sebesar 96.95%, terlihat pada gambar 4 hasil prediksi yang didapatkan sehingga mendapatkan akurasi tertinggi dari 10% data uji yang berjumlah 655 data yang diprediksi dengan jumlah data pada kondisi cuaca cerah berawan sebanyak 268 data, pada mendung/hujan ringan sebanyak 21 data, hujan sedang sebanyak 4 data, hujan lebat sebanyak 70 data, dan hujan sangat lebat sebanyak 292 data. Kolom true label adalah data aktual yang terjadi dari stasiun dan predict label adalah hasil prediksi yang dilakukan pada penelitian ini.

Hasil prediksi yang didapatkan penelitian saat menggunakan naïve bayes sebelum dioptimasi pada kondisi cerah berawan, mendung/hujan, hujan sedang dan hujan lebat hasil prediksi benar dengan data actual yang terjadi, namun saat terdapat data yang seharusnya kondisi hujan sangat lebat hasil prediksi sebelum dioptimasi terjadi 20 kesalahan prediksi, setelah dilakukan optimasi dengan menggunakan metode *Univariate Feature Selection* ANOVA-f test dan Hyperparameter Tuning GridsearchCV, hasil 20 kesalahan prediksi pada data hujan sangat lebat berkurang menjadi 7 kesalahan saat setelah dioptimasi.

**Gambar 6.** Confusion matrix Hasil Prediksi Gaussian Naïve Bayes Sebelum dan Sesudah di Optimasi

Pembahasan

Pengujian yang dilakukan pada penelitian ini dilakukan sebanyak 10 kali percobaan pada masing-masing data set. di pecah menjadi 5 kali percobaan sebelum dioptimasi dan 5 kali setelah dioptimasi dengan membagi data set menjadi data uji dan data latih 10%:90%, 20%:80%, 30%:70%, 40%:60% dan 50%:50%. Percobaan data set tanpa optimasi mendapatkan akurasi tertinggi 96.95% dengan menggunakan 90% data latih dan 10% data uji, setelah dioptimasi akurasi prediksi meningkat 1.98% menjadi 98.93% dengan data latih dan data uji yang sama 90%:10% pada percobaan optimasi ini menjadi akurasi tertinggi dari 10 percobaan yang dilakukan. Pada tabel 3 selisih peningkatan juga terjadi pada percobaan lainnya yaitu 20%:80%, 30%:70%, 40%:60% dan 50%:50%, dengan menggunakan metode *Univariate Feature Selection* ANOVA-f test dan *Hyperparameter Tuning GridsearchCV*, dengan hasil

prediksi seperti pada hasil confusion matrix gambar 4 dan gambar 5. Perbandingan seluruh nilai akurasi bisa dilihat pada gambar 6, tidak hanya performa *Accuracy* peningkatan juga terjadi dalam performa *Precision*, *Recall*, dan *F1-Score* pada model *Gaussian Naïve Bayes* sebelum dioptimasi dan sesudah dioptimasi ditunjukkan pada tabel 1 dan tabel 2.

Pada penelitian ini, hasil prediksi menggunakan *Naïve Bayes* sebelum dioptimasi untuk kondisi cuaca cerah berawan, mendung/hujan, hujan sedang, dan hujan lebat menunjukkan prediksi yang benar sesuai dengan data sebenarnya. Namun, saat ada data yang seharusnya merupakan kondisi hujan sangat lebat, terdapat 20 kesalahan prediksi sebelum dilakukan optimasi. Setelah melakukan optimasi dengan menggunakan metode *Univariate Feature Selection ANOVA-f test* dan *Hyperparameter Tuning GridsearchCV*, jumlah kesalahan prediksi pada data hujan sangat lebat berkurang menjadi 7 kesalahan. Perubahan ini dapat dilihat dalam gambar 4. Dengan demikian, hasil optimasi telah berhasil meningkatkan performa prediksi untuk kondisi hujan sangat lebat dan mengurangi jumlah kesalahan prediksi yang sebelumnya terjadi.

Tabel 4. Rekapitulasi Hasil Prediksi Dari 655 Data Prediksi

<i>Kondisi Cuaca</i>	<i>Hasil Prediksi</i>	<i>Data Aktual</i>	<i>Kesalahan Prediksi</i>
Cerah Berawan NB	268	268	0
Cerah Berawan OP_NB	268	268	0
Hujan Ringan NB	21	21	0
Hujan Ringan OP_NB	21	21	0
Hujan Sedang NB	4	4	0
Hujan Sedang OP_NB	4	4	0
Hujan Lebat NB	70	70	0
Hujan Lebat OP_NB	70	70	0
Hujan Sangat Lebat NB	272	292	20
Hujan Sangat Lebat OP_NB	285	292	7

Hasil temuan ini, *Gaussian Naive Bayes* memiliki kelebihan yang mampu menangani tugas klasifikasi dengan lebih dari dua kelas dan dapat diperluas dengan mudah untuk mengatasi masalah multikelas. (Widyassari & Suryani, 2021; Muhamad et al., 2017) . Meskipun memiliki beberapa kelebihan, *Gaussian Naive Bayes* juga memiliki beberapa kelemahan, seperti kesulitan dalam menangani ketergantungan antar fitur dan ketidakmampuannya memodelkan hubungan yang rumit antara fitur (Harianto et al., 2020). Metode *Hyperparameter tuning GridsearchCV* dan *Univariate Feature Selection ANOVA-f test* dalam model *Gaussian Naive Bayes* dapat menjadi solusi dan memberikan hasil yang baik dibandingkan dengan temuan sebelumnya (Yani et al., 2022), karena penelitian tersebut memiliki kendala yaitu banyak *variance* yang nilainya hampir sama untuk setiap kategori, sehingga sulit untuk dimodelkan bahkan setelah dilakukan optimasi oleh metode *ensemble* yang dilakukan hasil temuan penelitian sebelumnya menunjukkan akurasi sebesar 49.76% dengan dataset sebanyak 3652 data dalam rentang waktu 10 tahun. Namun, penelitian yang dilakukan ini menggunakan dataset yang lebih besar, yaitu sebanyak 6454 data dalam rentang waktu 20,5 tahun, dataset yang lebih banyak karena saran yang diberikan pada penelitian (Utami et al., 2021) yang memiliki hasil akurasi 70% menggunakan metode LSH dan KNN.

Hasil penelitian yang dilakukan ini menunjukkan bahwa akurasi prediksi cuaca meningkat menjadi 98.93%, lebih tinggi dari penelitian sebelumnya. Dengan membandingkan hasil ini dengan temuan sebelumnya, penelitian yang dilakukan ini menunjukkan adanya peningkatan akurasi yang signifikan dan perluasan cakupan dataset, yang mengindikasikan kebaruan dan kontribusi penelitian dalam bidang prediksi cuaca menggunakan model

Gaussian Naïve Bayes dioptimasi dengan metode *Hyperparameter Tuning GridsearchCV* dan *Univariate Feature Selection ANOVA-f test*, sehingga penelitian yang dilakukan ini berhasil meningkatkan performa model dan memberikan pemahaman yang lebih luas tentang prediksi cuaca dengan pendekatan *machine learning*.

SIMPULAN

Penggunaan teknik optimasi *Hyperparameter Tuning GridsearchCV* dan *Univariate Feature Selection ANOVA-f test* terbukti efektif dalam meningkatkan akurasi prediksi cuaca pada model *Gaussian Naive Bayes*. Temuan ini menunjukkan hasil prediksi saat menggunakan *Naïve Bayes* sebelum dioptimasi pada kondisi cerah berawan, mendung/hujan, hujan sedang dan hujan lebat mendapat hasil prediksi benar dengan data aktual yang terjadi. Namun saat terdapat terjadi 20 kesalahan prediksi pada kondisi cuaca hujan sangat lebat, setelah dilakukan optimasi dengan menggunakan metode *Univariate Feature Selection ANOVA-f test* dan *Hyperparameter Tuning GridsearchCV*, hasil 20 kesalahan prediksi pada data hujan sangat lebat berkurang menjadi 7 kesalahan saat setelah dioptimasi. Penggunaan *Hyperparameter Tuning* membantu mencari kombinasi parameter yang optimal untuk model *Gaussian Naive Bayes*. Selain itu, fitur-fitur yang tidak relevan dapat dieliminasi, sehingga model hanya mempertimbangkan fitur-fitur yang berkontribusi signifikan dalam melakukan prediksi cuaca.

REFERENSI

- Adlini, Anisya H.D, Sarah Y, Octavia C, & Sauda J.M. (2022). Metode Penelitian Kualitatif Studi Pustaka. *Edumaspul: Jurnal Pendidikan*, 6(1), 974–980. <https://doi.org/10.33487/edumaspul.v6i1.3394>
- Aguni, L., Chabaa, S., Ibnyaich, S., & Zeroual, A. (2021). Predicting the notch band frequency of an ultra-wideband antenna using artificial neural networks. *Telkomnika (Telecommunication Computing Electronics and Control)*, 19(1), 1–8. <https://doi.org/10.12928/TELKOMNIKA.V19I1.15912>
- Alhakeem, Z. M., Jebur, Y. M., Henedy, S. N., Imran, H., Bernardo, L. F. A., & Hussein, H. M. (2022). Prediction of Ecofriendly Concrete Compressive Strength Using Gradient Boosting Regression Tree Combined with GridSearchCV Hyperparameter-Optimization Techniques. *Materials*, 15(21). <https://doi.org/10.3390/ma15217432>
- Hariato, H., Sunyoto, A., & Sudarmawan, S. (2020). Optimasi Algoritma Naïve Bayes Classifier untuk Mendeteksi Anomaly dengan Univariate Fitur Selection. *Edumatic: Jurnal Pendidikan Informatika*, 4(2), 40–49. <https://doi.org/10.29408/edumatic.v4i2.2433>
- Harpale, V., & Bairagi, V. (2021). An adaptive method for feature selection and extraction for classification of epileptic EEG signal in significant states. *Journal of King Saud University - Computer and Information Sciences*, 33(6), 668–676. <https://doi.org/10.1016/j.jksuci.2018.04.014>
- Intan, I., Aminah Dinayati Ghani, S., Koswara, A. T., Dipa Makassar, U., Arsip Nasional Republik Indonesia, K., & Kemerdekaan, J. P. (2021). Analisis Performansi Prakiraan Cuaca Menggunakan Algoritma Machine Learning Performance Analysis of Weather Forecasting using Machine Learning Algorithms. *Jurnal_Pekommas_Vol._6_No, 2*, 1–8. <https://doi.org/10.30818/jpkm.2021.2060221>
- Jebadurai, I. J., Paulraj, G. J. L., Jebadurai, J., & Silas, S. (2022). Experimental Analysis of Filtering-Based Feature Selection Techniques for Fetal Health Classification. *Serbian Journal of Electrical Engineering*, 19(2), 207–224. <https://doi.org/10.2298/SJEE2202207J>

- Muhamad, H., Prasajo, C. A., Sugianto, N. A., Surtiningsih, L., & Cholissodin, I. (2017). *Optimasi Naïve Bayes Classifier Dengan Menggunakan Particle Swarm Optimization Pada Data Iris*. 4(3), 180–184. <http://dx.doi.org/10.25126/jtiik.201743251>
- Maisat, & Ashafidz F.D. (2023). Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM. *Online) Teknologi: Jurnal Ilmiah Sistem Informasi*, 13(1), 8–15. <https://doi.org/10.26594/teknologi.v13i1.3098>
- Oshodi. (2022). Machine Learning-based Algorithms for Weather Forecasting. *Preprints*, 1(1), 1–6. <https://doi.org/10.20944/preprints202206.0428.v1>
- Rangkuti, Muhamamd V. A, & Wawan G. (2021). Penerapan Algoritma K-Nearest Neighbor (KNN) Dalam Memprediksi Dan Menghitung Tingkat Akurasi Data Cuaca Di Indonesia. *HEXAGON*, 2(2), 11–16. <https://doi.org/10.36761/hexagon.v2i2.1082>
- Sari, V., Firdausi, F., & Azhar, Y. (2020). Perbandingan Prediksi Kualitas Kopi Arabika dengan Menggunakan Algoritma SGD, Random Forest dan Naive Bayes. *Edumatic: Jurnal Pendidikan Informatika*, 4(2), 1–9. <https://doi.org/10.29408/edumatic.v4i2.2202>
- Shakeela, S., Shankar, N. S., Reddy, P. M., Tulasi, T. K., & Koneru, M. M. (2021). Optimal ensemble learning based on distinctive feature selection by univariate ANOVA-F statistics for IDS. *International Journal of Electronics and Telecommunications*, 67(2), 267–275. <https://doi.org/10.24425/ijet.2021.135975>
- Siregar. (2020). Klasifikasi Untuk Prediksi Cuaca Menggunakan Esemble Learning. *PETIR*, 13(2), 138–147. <https://doi.org/10.33322/petir.v13i2.998>
- Siregar, Sutan F, Yana C, & Bayu P. (2020). Perbandingan Algoritme Klasifikasi Untuk Prediksi Cuaca. *Jurnal Accounting Information System (AIMS)*, 3(1). <https://journal.stkom.ac.id/index.php/aims/article/view/92>
- Sunarmi, N., Kumailia, E. N., Nurfaiza, N., Nikmah, A. K., Aisyah, H. N., Sriwahyuni, I., & Lailly, S. N. (2022). Analisis Faktor Unsur Cuaca terhadap Perubahan Iklim Di Kabupaten Pasuruan pada Tahun 2021 dengan Metode Principal Component Analysis. *Newton-Maxwell Journal of Physics*, 3(2), 56–64. <https://doi.org/10.33369/nmj.v3i2.23380>
- Susanti, S., Sari, A. A., Anam, M. K., Jamaris, M., & Hamdani, H. (2022). Sistem Prediksi Keuntungan Influencer Pengguna E-Commerce Shopee Affiliates menggunakan Metode Naïve Bayes. *Edumatic: Jurnal Pendidikan Informatika*, 6(2), 394–403. <https://doi.org/10.29408/edumatic.v6i2.6787>
- Susatyono. (2021). *KECERDASAN BUATAN, Kajian Konsep dan Penerapan: Vol. 7(1)*. Yayasan Prima Agus Teknik.
- Utami, Rini, D. P., & Lestari, E. (2021). Prediksi Cuaca di Kota Palembang Berbasis Supervised Learning Menggunakan Algoritma K-Nearest Neighbour. *JUPITER: Jurnal Penelitian Ilmu Dan Teknologi Komputer*, 13(1), 09–18. <https://jurnal.polsri.ac.id/index.php/jupiter/article/view/3243>
- Widyassari, A. P., & Suryani, P. E. (2021). Komparasi Metode Naïve Bayes dan SAW untuk Pemilihan Penerimaan Insentif Karyawan. *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, 3(02), 149–159. <https://doi.org/10.46772/intech.v3i02.555>
- Yani, Aradea, & Husni. (2022). Optimizing Weather Forecast Using Ensemble Method on Naïve Bayes and C4.5. *JuTISI*, 8(3), 607–619. <https://doi.org/10.28932/jutisi.v8i3.5455>

LAMPIRAN SOURCE CODE TUGAS AKHIR

CODINGAN KLASIFIKASI PREDIKSI NAIVE BAYES SEBELUM DIOPTIMASI

```
#IMPORT LIBRARY YANG DIBUTUHKAN
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, classification_report
import numpy as np
from IPython.display import display
import matplotlib.pyplot as plt

#PRE PROCESSING DATA SEBELUM DIOPTIMASI

data = pd.read_excel('DATA_CUACA1.xlsx') #Membaca Dataset Yang
Telah Dikumpulkan
data.info()
display(data) # Melihat Data Mentah
print ("\n")
display(data.describe()) #Melihat Keterangan Data Mentah
data = data.drop(['Tanggal', 'AAT'], axis=1) # Menghapus kolom yang
tidak diperlukan
data = data.dropna() # Menghapus baris dengan nilai kosong (NaN)
data = data.drop_duplicates() # Menghapus baris dengan nilai
duplikat
display (data) # Membaca Data Yang Telah dilakukan pre processing
display (data.describe()) # Membaca Keterangan data yang telah
dilakukan Pre Processing
display(data.info())

# Ganti "nama_fitur" dengan nama kolom yang ingin Anda tampilkan
kelasnya
nama_fitur = "Suhu Min"

# Menampilkan nilai tertinggi dan terendah pada kelas fitur
nilai_tertinggi = data[nama_fitur].max()
nilai_terendah = data[nama_fitur].min()

print("Nilai tertinggi pada fitur", nama_fitur, ":",
nilai_tertinggi)
print("Nilai terendah pada fitur", nama_fitur, ":",
nilai_terendah)
```

```

# Memisahkan fitur dan target
X = data.drop('Kondisi Cuaca', axis=1) # X dinyatakan sebagai
fitur
y = data['Kondisi Cuaca'] # y dinyatakan sebagai target

#KLASIFIKASI PREDIKSI MODEL GAUSSIAN NAIVE BAYES

results = [] # Membuat Daftar Untuk melakukan beberapa percobaan
model

test_sizes = [0.1,0.2,0.3,0.4,0.5] # Memasukkan daftar 5 uji
percobaan yang akan dilakukan
for test_size in test_sizes:
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=test_size, random_state=0) # Bagi data menjadi set
pelatihan dan set pengujian

    # Inisialisasi dan latih model Naive Bayes
    model = GaussianNB()
    model.fit(X_train, y_train)

    # Lakukan prediksi pada set pengujian
    y_pred = model.predict(X_test)

    # Hitung metrik evaluasi
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test,
y_pred,average='macro',zero_division=0)
    recall = recall_score(y_test, y_pred,
average='macro',zero_division=0)
    f1 = f1_score(y_test, y_pred, average='macro',zero_division=0)

    # Simpan hasil ke lis
    results.append((test_size, accuracy, precision, recall, f1))

# Membaca Hail List dari 5 percobaan model sebelum dioptimasi
for test_size, accuracy, precision, recall, f1 in results:
    print(f"\033[1mTest Size: {test_size:}\033[0m")
    print(f"Accuracy: {accuracy*100:.2f}%, Precision:
{precision*100:.2f}%, Recall: {recall*100:.2f}%, F1-score:
{f1*100:.2f}%\n")

```

DIAGRAM HASIL AKURASI PREDIKSI SEBELUM DIOPTIMASI DENGAN 5 KALI PERCOBAAN

```
[  
# Hasil evaluasi  
results = [  
    ("90%:10%", 96.95),  
    ("80%:20%", 96.10),  
    ("70%:30%", 95.42),  
    ("60%:40%", 95.53),  
    ("50%:50%", 95.36)  
]  
  
# Ekstrak data dari hasil evaluasi  
test_sizes, accuracy_scores = zip(*results)  
  
# Daftar label untuk sumbu y  
y_labels = ['100.00%', '90.00%', '98.00%', '97.00%', '96.00%', '95.00%',  
            '94.00%', '93.00%', '92.00%']  
  
# Buat diagram grafik garis  
plt.figure(figsize=(7, 4))  
plt.plot(test_sizes, accuracy_scores, linestyle='--', marker='o', label='Accuracy', color='blue', markerfacecolor='black')  
  
plt.xlabel('')  
plt.ylabel('')  
plt.title('')  
  
# Menambahkan keterangan nilai hasil di atas garis  
for i, acc in enumerate(accuracy_scores):  
    plt.annotate(f'{acc:.2f}%', (test_sizes[i], accuracy_scores[i]), color='blue', textcoords="offset points", xytext=(7, 7), ha='center', va='bottom')  
  
plt.yticks([100.00, 99.00, 98.00, 97.00, 96.00, 95.00, 94.00, 93.00, 92.00], y_labels)  
  
plt.grid(True, linestyle='--', alpha=0.7, axis='y') # Hanya menambahkan grid pada sumbu y  
  
# Menambahkan keterangan garis pada legend  
plt.legend(loc='upper left', bbox_to_anchor=(1.03, 0.6), handlelength=3, handletextpad=1)  
  
plt.show() # Menampilkan Hasil Grafik
```

DIAGRAM AKURASI,PRESISI,RECALL,F1_SCORE SEBELUM DIOPTIMASI DENGAN 5 KALI PERCOBAAN

```
# Hasil evaluasi
results = [
    ("90%:10%", 96.95),
    ("80%:20%", 96.10),
    ("70%:30%", 95.42),
    ("60%:40%", 95.53),
    ("50%:50%", 95.36)
]

# Ekstrak data dari hasil evaluasi
test_sizes, accuracy_scores = zip(*results)

# Daftar label untuk sumbu y
y_labels =
['100.00%', '99.00%', '98.00%', '97.00%', '96.00%', '95.00%', '94.00%', '93.00%', '92.00%', '91.00%', '90.00%', '89.00%', '88.00%', '87.00%', '86.00%', '85.00%']

plt.figure(figsize=(8, 5))

# Buat diagram grafik garis accuracy_score
plt.plot(test_sizes, accuracy_scores, linestyle='--', marker='o',
label='Accuracy_Score', color='blue', markerfacecolor='black')

# Menambahkan garis plot kedua precision_score
precision_scores = [95.56, 94.05, 87.98, 87.67, 86.55]
plt.plot(test_sizes, precision_scores, linestyle='--', marker='o',
label='Precision_Score', color='red', markerfacecolor='black')

# Menambahkan garis plot ketiga recall_score
recall_scores = [98.63, 98.13, 91.96, 92.89, 92.53]
plt.plot(test_sizes, recall_scores, linestyle='--', marker='o',
label='Recall_Score', color='green', markerfacecolor='black')

# Menambahkan garis plot keempat f1_score
f1_scores = [96.79, 95.63, 89.37, 89.79, 88.85]
plt.plot(test_sizes, f1_scores, linestyle='--', marker='o',
label='F1_Score', color='purple', markerfacecolor='black')
plt.xlabel('')
plt.ylabel('')
plt.title('')
```

```

# Menambahkan keterangan nilai hasil di atas garis untuk
accuracy_score
for i, acc in enumerate(accuracy_scores):
    plt.annotate(f'{acc:.2f}%', (test_sizes[i],
accuracy_scores[i]), color='blue',textcoords="offset points",
xytext=(1, 5), ha='center', va='bottom')

# Menambahkan keterangan nilai hasil di atas garis untuk
precision_score
for i, prec in enumerate(precision_scores):
    plt.annotate(f'{prec:.2f}%', (test_sizes[i],
precision_scores[i]), color='red',textcoords="offset points",
xytext=(-0, -20), ha='center', va='bottom')

# Menambahkan keterangan nilai hasil di atas garis untuk
recall_score
for i, rec in enumerate(recall_scores):
    plt.annotate(f'{rec:.2f}%', (test_sizes[i], recall_scores[i]),
color='green', textcoords="offset points", xytext=(-1, -15),
ha='center', va='bottom')

# Menambahkan keterangan nilai hasil di atas garis untuk f1_score
for i, f1 in enumerate(f1_scores):
    plt.annotate(f'{f1:.2f}%', (test_sizes[i],
f1_scores[i]),color='purple', textcoords="offset points",
xytext=(1, -13), ha='center', va='center')

plt.yticks([100.00,
99.00,98.00,97.00,96.00,95.00,94.00,93.00,92.00,91.00,90.00,89.00,
88.00,87.00,86.00,85.00], y_labels)
plt.grid(True, linestyle='--', alpha=0.7, axis='y') # Hanya
menambahkan grid pada sumbu y

# Menambahkan keterangan garis pada legend
plt.legend(loc='upper left', bbox_to_anchor=(1.03, 0.6),
handlelength=3, handletextpad=1)

plt.show()

```

CLASSIFICATION REPORT MODEL SEBELUM DIOPTIMASI

[]

```

results = [] # Membuat Daftar Untuk melakukan beberapa percobaan model
test_sizes = [0.1, 0.2, 0.3, 0.4, 0.5] # Memasukkan daftar 5 uji percobaan yang akan dilakukan
for test_size in test_sizes:
    # Bagi data menjadi set pelatihan dan set pengujian
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=0)
    # Inisialisasi dan latih model Naive Bayes
    model = GaussianNB()
    model.fit(X_train, y_train)
    # Lakukan prediksi pada set pengujian
    y_pred = model.predict(X_test)
    # Get the classification report
    report = classification_report(y_test, y_pred)
    # Simpan hasil ke list
    results.append((test_size, report))
for i, (test_size, report) in enumerate(results):
    print(f"\033[1mUJI PERCOBAAN {i + 1} - Test Size: {test_size:.2f}\033[0m")
    print(f"Classification Report:\n{report}\n")

```

CONFUSION MATRIX MODEL SEBELUM OPTIMASI

```

[]

import matplotlib.pyplot as plt
import seaborn as sns

cm = confusion_matrix(y_test, y_pred)

# List to store the confusion matrix for all test sizes
confusion_matrices = []

test_sizes = [0.1, 0.2, 0.3, 0.4, 0.5] # Memasukkan daftar 5 uji percobaan yang akan dilakukan
for test_size in test_sizes:

```

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test
_size=test_size, random_state=0) # Bagi data menjadi set pelatiha
n dan set pengujian

# Inisialisasi dan latih model Naive Bayes

model = GaussianNB()

model.fit(X_train, y_train)

# Lakukan prediksi pada set pengujian

y_pred = model.predict(X_test)

# Hitung metrik evaluasi

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred, average='macro', zer
o_division=0)

recall = recall_score(y_test, y_pred, average='macro', zero_div
ision=0)

f1 = f1_score(y_test, y_pred, average='macro', zero_division=0)

# Simpan confusion matrix ke list

confusion_matrices.append(confusion_matrix(y_test, y_pred))

# Print the results for each test size

print(f"\033[1mTest Size: {test_size}\033[0m")

print(f"Accuracy: {accuracy*100:.2f}%, Precision: {precision*1
00:.2f}%, Recall: {recall*100:.2f}%, F1-score: {f1*100:.2f}%")

print("Confusion Matrix:")

print(confusion_matrices[-1])

print("\n")

# Define the class labels

class_labels = ['Cerah Berawan', 'Mendung/Hujan Ringan', 'Hujan Se
dang', 'Hujan Lebat', 'Hujan Sangat Lebat']

# Plot the confusion matrices for all test sizes

for idx, test_size in enumerate(test_sizes):

    plt.figure(figsize=(5, 4))

```

```

sns.heatmap(confusion_matrices[idx], annot=True, cmap='Blues',
fmt='g')

plt.title(f"Confusion Matrix NB (Test Size: {test_size})")

plt.xticks(ticks=[0.5, 1.5, 2.5, 3.5, 4.5], labels=class_labels, rotation=45, ha='right')

plt.yticks(ticks=[0.5, 1.5, 2.5, 3.5, 4.5], labels=class_labels, rotation=0, va='center')

# Add "True Label" and "Predicted Label" outside the plot

plt.figtext(-
0.35, 0.5, "True Label", fontsize=12, rotation=90,color='gray', ha
='left', va='center')

plt.figtext(0.5, 0.3, "Predicted Label", fontsize=12,color='gray',rotation=-360, ha='center', va='center')

plt.show()

plt.show()

```

OPTIMASI MODEL NAIVE BAYES MENGGUNAKAN HYPERPARAMETER TUNING GRIDSEARCHCV DAN UNIVARIATE FEATURE SELECTION ANOVA-F TEST

```

[]
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import SelectKBest,f_classif
#Optimasi pada pengolahan data
# Mengatasi nilai negatif dengan Min-Max scaling
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
import pandas as pd
X_scaled_df = pd.DataFrame(X_scaled, columns=X.columns)
display(X_scaled_df)
# Pemilihan fitur
selector = SelectKBest(score_func=f_classif, k=6)
X_selected = selector.fit_transform(X_scaled, y)
# List to store the results of each experiment
results = []
# Perform 5 experiments with different test sizes

```

```

test_sizes2 = [0.1, 0.2, 0.3, 0.4, 0.5]
for test_size2 in test_sizes2:
    # Bagi data menjadi set pelatihan dan set pengujian
    X_train2, X_test2, y_train2, y_test2 = train_test_split(X_selected, y, test_size=test_size2, random_state=0)
    # Inisialisasi dan latih model Naive Bayes
    model = GaussianNB()
    # Optimasi Parameter dengan Grid Search
    param_grid = {'var_smoothing': [1e-9, 1e-8, 1e-7]}
    grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=10)
    grid_search.fit(X_train2, y_train2)
    # Mendapatkan model terbaik dari Grid Search
    best_model = grid_search.best_estimator_
    # Lakukan prediksi pada set pengujian
    y_pred2 = best_model.predict(X_test2)
    # Hitung metrik evaluasi
    accuracy2 = accuracy_score(y_test2, y_pred2)
    precision2 = precision_score(y_test2, y_pred2, average='macro', zero_division=0)
    recall2 = recall_score(y_test2, y_pred2, average='macro', zero_division=0)
    f12 = f1_score(y_test2, y_pred2, average='macro', zero_division=0)
    # Simpan hasil ke list
    results.append((test_size2, accuracy2, precision2, recall2, f12))
# Print the results
for test_size2, accuracy, precision, recall, f1 in results:
    print(f"\033[1mTest Size: {test_size2}\033[0m")
    print(f"Accuracy: {accuracy*100:.2f}%, Precision: {precision*100:.2f}%, Recall: {recall*100:.2f}%, F1-score: {f1*100:.2f}%\n")

```

CLASSIFICATION REPORT SETELAH DIOPTIMASI MODEL GAUSSIAN NAIVE BAYES

```
[]

results = [] # Membuat Daftar Untuk melakukan beberapa percobaan model

# Perform 5 experiments with different test sizes
test_sizes2 = [0.1, 0.2, 0.3, 0.4, 0.5]

for test_size2 in test_sizes2:

    # Bagi data menjadi set pelatihan dan set pengujian
    X_train2, X_test2, y_train2, y_test2 = train_test_split(X_selected, y, test_size=test_size2, random_state=0)

    # Inisialisasi dan latih model Naive Bayes
    model = GaussianNB()

    # Optimasi Parameter dengan Grid Search
    param_grid = {'var_smoothing': [1e-9, 1e-8, 1e-7]}

    grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=10)

    grid_search.fit(X_train2, y_train2)

    # Mendapatkan model terbaik dari Grid Search
    best_model = grid_search.best_estimator_

    # Lakukan prediksi pada set pengujian
    y_pred2 = best_model.predict(X_test2)

    # Get the classification report
    report = classification_report(y_test2, y_pred2)

    # Simpan hasil ke list
    results.append((test_size2, report))

for i, (test_size, report) in enumerate(results):

    print(f"\033[1mUJI PERCOBAAN {i + 1} - Test Size: {test_size:.2f}\033[0m")

    print(f"Classification Report:\n{report}\n")
```

DIAGRAM AKURASI OPTIMASI MODEL GAUSSIAN NAIVE BAYES

```
[]

# Hasil evaluasi

results = [

    ("90%:10%", 98.93),

    ("80%:20%", 98.70),

    ("70%:30%", 98.17),

    ("60%:40%", 98.13),

    ("50%:50%", 98.04)

]

# Ekstrak data dari hasil evaluasi

test_sizes, accuracy_scores = zip(*results)

# Daftar label untuk sumbu y

# Daftar label untuk sumbu y

y_labels = ['100.00%', '99.00%', '98.00%', '97.00%']

# Buat diagram grafik garis

plt.figure(figsize=(8, 4))

plt.plot(test_sizes, accuracy_scores, linestyle='--',
         marker='o', label='Accuracy', color='blue', markerfacecolor='black')

plt.xlabel('')

plt.ylabel('')

plt.title('')

# Menambahkan keterangan nilai hasil di atas garis

for i, acc in enumerate(accuracy_scores):

    plt.annotate(f'{acc:.2f}%', (test_sizes[i], accuracy_scores[i]),
                color='blue', textcoords="offset points", xytext=(0, 7), ha='center', va='bottom')

plt.yticks([100.00, 99.00, 98.00, 97.00], y_labels)
```

```
plt.grid(True, linestyle='--
', alpha=0.7, axis='y') # Hanya menambahkan grid pada sumbu y

# Menambahkan keterangan garis pada legend

plt.legend(loc='upper left', bbox_to_anchor=(1.03, 0.6), handlelen
gth=3, handletextpad=1)

plt.show() # Menampilkan Hasil Grafik
```

DIAGRAM AKURASI,PRESISI,RECALL,F1_SCORE DENGAN 5 KALI PERCOBAAN SETELAH MODEL DI OPTIMASI

```
[]

import matplotlib.pyplot as plt

# Hasil evaluasi

results = [

    ("90%:10%", 98.93),

    ("80%:20%", 98.70),

    ("70%:30%", 98.17),

    ("60%:40%", 98.13),

    ("50%:50%", 98.04)

]

# Ekstrak data dari hasil evaluasi

test_sizes, accuracy_scores = zip(*results)

# Daftar label untuk sumbu y

y_labels = ['100.00%', '99.50%', '99.00%', '98.50%', '98.00%', '97.50%',
', '97.00%', '96.50%', '96.00%', '95.50%', '95.00%', '94.50%', '94.00%', '
93.50%', '93.00%', '92.50%', '92.00%', '91.50%', '91.00%']

plt.figure(figsize=(10, 7))

# Buat diagram grafik garis accuracy_score

plt.plot(test_sizes, accuracy_scores, linestyle='--
', marker='o', label='Accuracy_Score', color='blue', markerfacecol
or='black')

# Menambahkan garis plot kedua precision_score
```

```

precision_scores = [98.18, 97.38, 93.32, 92.21, 92.10]

plt.plot(test_sizes, precision_scores, linestyle='--',
         marker='o', label='Precision_Score', color='red', markerfacecolor='black')

# Menambahkan garis plot ketiga recall_score

recall_scores = [99.52, 99.30, 93.50, 94.36, 93.96]

plt.plot(test_sizes, recall_scores, linestyle='--',
         marker='o', label='Recall_Score', color='green', markerfacecolor='black')

# Menambahkan garis plot keempat f1_score

f1_scores = [98.80, 98.28, 93.25, 93.21, 92.97]

plt.plot(test_sizes, f1_scores, linestyle='--',
         marker='o', label='F1_Score', color='purple', markerfacecolor='black')

plt.xlabel('')
plt.ylabel('')
plt.title('')

# Menambahkan keterangan nilai hasil di atas garis untuk accuracy_score

for i, acc in enumerate(accuracy_scores):
    plt.annotate(f'{acc:.2f}%', (test_sizes[i], accuracy_scores[i]),
                color='blue', textcoords="offset points", xytext=(7, 7), ha='center', va='bottom')

# Menambahkan keterangan nilai hasil di atas garis untuk precision_score

for i, prec in enumerate(precision_scores):
    plt.annotate(f'{prec:.2f}%', (test_sizes[i], precision_scores[i]),
                color='red', textcoords="offset points", xytext=(-1, -27), ha='center', va='bottom')

# Menambahkan keterangan nilai hasil di atas garis untuk recall_score

for i, rec in enumerate(recall_scores):

```

```

plt.annotate(f'{rec:.2f}%', (test_sizes[i], recall_scores[i]),
color='green',textcoords="offset points", xytext=(8, 8), ha='center', va='bottom')

# Menambahkan keterangan nilai hasil di atas garis untuk f1_score
for i, f1 in enumerate(f1_scores):
    plt.annotate(f'{f1:.2f}%', (test_sizes[i], f1_scores[i]),color
='purple', textcoords="offset points", xytext=(20, -
15), ha='right', va='bottom')

plt.yticks([100.00,99.50,99.00,98.50, 98.00,97.50,97.00,96.50,96.00,
95.50,95.00,94.50,94.00,93.50,93.00,92.50,92.00,91.50,91.00], y_labels)

plt.grid(True, linestyle='--', alpha=0.7, axis='y') # Hanya menambahkan grid pada sumbu y

# Menambahkan keterangan garis pada legend

plt.legend(loc='upper left', bbox_to_anchor=(1.03, 0.6), handlelength=3, handletextpad=1)

plt.show()

```

MENDAPATKAN SKOR FITUR TERBAIK MENGGUNAKAN UNIVARIATE FEATURE SELECTION ANOVA-F TEST

```

[]

# Mendapatkan skor fitur dari SelectKBest

feature_scores = selector.scores_

# Membuat DataFrame untuk menyimpan skor masing-masing fitur

df_scores = pd.DataFrame({"Feature": X_train.columns, "Score": feature_scores})

# Menampilkan Dataframe

display(df_scores)

# Mendapatkan fitur terbaik yang dipilih

selected_features = X_train.columns[selector.get_support()]

print("Fitur Terbaik yang dipilih sebagai optimasi:")

print(selected_features)

```

CONFUSION MATRIX MODEL GAUSSIAN NAIVE BAYES SETELAH DI OPTIMASI PADA PENGUJIAN DATASET

```
[]

import matplotlib.pyplot as plt

import seaborn as sns

# List to store the confusion matrix for all test sizes
confusion_matrices = []

# Perform 5 experiments with different test sizes
test_sizes2 = [0.1, 0.2, 0.3, 0.4, 0.5]

for test_size2 in test_sizes2:

    # Bagi data menjadi set pelatihan dan set pengujian
    X_train2, X_test2, y_train2, y_test2 = train_test_split(X_selected, y, test_size=test_size2, random_state=0)

    # Inisialisasi dan latih model Naive Bayes
    model = GaussianNB()

    # Optimasi Parameter dengan Grid Search
    param_grid = {'var_smoothing': [1e-9, 1e-8, 1e-7]}

    grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=10)

    grid_search.fit(X_train2, y_train2)

    # Mendapatkan model terbaik dari Grid Search
    best_model = grid_search.best_estimator_

    # Lakukan prediksi pada set pengujian
    y_pred2 = best_model.predict(X_test2)

    # Hitung metrik evaluasi
    accuracy2 = accuracy_score(y_test2, y_pred2)

    precision2 = precision_score(y_test2, y_pred2, average='macro', zero_division=0)

    recall2 = recall_score(y_test2, y_pred2, average='macro', zero_division=0)
```

```

f12 = f1_score(y_test2, y_pred2, average='macro', zero_divisio
n=0)

# Simpan confusion matrix ke list
confusion_matrices.append(confusion_matrix(y_test2, y_pred2))

# Print the results for each test size
print(f"\033[1mTest Size: {test_size2}\033[0m")

print(f"Accuracy: {accuracy2*100:.2f}%, Precision: {precision2
*100:.2f}%, Recall: {recall2*100:.2f}%, F1-score: {f12*100:.2f}%")

print("Confusion Matrix:")

print(confusion_matrices[-1])

print("\n")

# Define the class labels
class_labels = ['Cerah Berawan', 'Mendung/Hujan Ringan', 'Hujan Se
dang', 'Hujan Lebat', 'Hujan Sangat Lebat']

# Plot the confusion matrices for all test sizes
for idx, test_size2 in enumerate(test_sizes2):

    plt.figure(figsize=(5, 4))

    sns.heatmap(confusion_matrices[idx], annot=True, cmap='Blues',
fmt='g')

    plt.title(f"Confusion Matrix OP_NB (Test Size: {test_size2})")

    plt.xticks(ticks=[0.5, 1.5, 2.5, 3.5, 4.5], labels=class_label
s, rotation=45, ha='right')

    plt.yticks(ticks=[0.5, 1.5, 2.5, 3.5, 4.5], labels=class_label
s, rotation=0, va='center')

    # Add "True Label" and "Predicted Label" outside the plot

    plt.figtext(-
0.35, 0.5, "True Label", fontsize=12, rotation=90,color='gray', ha
='left', va='center')

    plt.figtext(0.5, -
0.3, "Predicted Label", fontsize=12,color='gray',rotation=-
360, ha='center', va='center')

    plt.show()

```

PERBANDINGAN 5 AKURASI UJI TEST DATASET PREDIKSI MODEL GAUSSIAN NAIVE BAYES SEBELUM DAN SESUDAH OPTIMASI

```
[  
  
import matplotlib.pyplot as plt  
  
# Hasil evaluasi  
results = [  
    ("NB;90%:10%", 96.95),  
    ("OP_NB;90%:10%", 98.93),  
    ("NB;80%:20%", 96.10),  
    ("OP_NB;80%:20%", 98.17),  
    ("NB;70%:30%", 95.42),  
    ("OP_NB;70%:30%", 98.17),  
    ("NB;60%:40%", 95.53),  
    ("OP_NB;60%:40%", 98.13),  
    ("NB;50%:50%", 95.36),  
    ("OP_NB;50%:50%", 98.04)  
]  
  
# Ekstrak data dari hasil evaluasi dan ubah formatnya  
test_sizes, accuracy_scores = zip(*[(size.replace(';', '\n'), acc)  
    for size, acc in results])  
  
# Daftar label untuk sumbu y  
y_labels = ['100.00%', '99.00%', '98.00%', '97.00%', '96.00%', '95.  
    .00%', '94.00%']  
  
# Tentukan dua warna untuk dua grup (NB dan OP_NB)  
colors = ['LIGHTBLUE', 'darkblue']  
  
# Buat diagram grafik batang  
plt.figure(figsize=(12, 4))  
  
# Gambar dua grup bar dengan warna yang berbeda  
bars = plt.bar(test_sizes, accuracy_scores, color=colors, alpha=1)
```

```

plt.xlabel('')
plt.ylabel('')
plt.title('')

# Menambahkan keterangan nilai hasil di atas batang untuk setiap grup
for bar, acc in zip(bars, accuracy_scores):
    plt.annotate(f'{acc:.2f}%', (bar.get_x() + bar.get_width() / 2, acc), color='BLACK', textcoords="offset points",
                 xytext=(0, 5), ha='center', va='bottom')

plt.ylim(94, 100)

plt.yticks([100.00, 99.00, 98.00, 97.00, 96.00, 95.00, 94.00], y_labels)

# Menambahkan keterangan di samping grafik
plt.text(10, 97, 'NB ', color='LIGHTBLUE', fontsize=12)
plt.text(10, 96.5, 'OPTIMASI NB', color='darkblue', fontsize=12)
plt.show() # Menampilkan Hasil Grafik

```

PENGUJIAN DATA UJI BARU

PENGUJIAN DATA UJI BARU SEBELUM DIOPTIMASI

```

[]

import pandas as pd

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report, confusion_matrix

# Baca dataset dengan data cuaca yang diketahui (data training)
data1 = pd.read_excel('DATA_CUACA1.xlsx')

# Menghapus kolom yang tidak diperlukan

data1 = data1.drop(['Tanggal', 'AAT'], axis=1) # Menghapus kolom yang tidak diperlukan

# Menghapus baris dengan nilai kosong (NaN)

```

```

data1 = data1.dropna()

data1 = data1.drop_duplicates()

display(data1)

# Memisahkan fitur dan target
X_train = data1.drop('Kondisi Cuaca', axis=1)
y_train = data1['Kondisi Cuaca']

# Baca dataset dengan data cuaca yang ingin diprediksi (data testing)
data2 = pd.read_excel('DATA_PRED.xlsx')
data2 = data2.drop(['Tanggal', 'AAT'], axis=1)
display(data2)
X_test = data2.drop('Kondisi Cuaca', axis=1)

# Baca dataset dengan data cuaca aktual (data aktual)
data3 = pd.read_excel('DATA_AKTUAL.xlsx')
display(data3)

y_actual = data3['Kondisi Cuaca'] # Data cuaca aktual untuk evaluasi

# Ganti "nama_fitur" dengan nama kolom yang ingin Anda tampilkan ke kelasnya
nama_fitur = "Suhu Min"

# Menampilkan nilai tertinggi dan terendah pada kelas fitur
nilai_tertinggi = data3[nama_fitur].max()
nilai_terendah = data3[nama_fitur].min()

print("Nilai tertinggi pada fitur", nama_fitur, ":", nilai_tertinggi)

print("Nilai terendah pada fitur", nama_fitur, ":", nilai_terendah)

# Inisialisasi dan latih model Naive Bayes
model = GaussianNB()
model.fit(X_train, y_train)

# Lakukan prediksi pada data cuaca yang ingin diprediksi

```

```

y_pred = model.predict(X_test)

# Tampilkan hasil prediksi
print("Prediksi Cuaca:")

print(y_pred)

# Tampilkan data cuaca aktual
print("Data Cuaca Aktual:")

print(y_actual)

# Bandingkan hasil prediksi dengan data cuaca aktual

hasil_perbandingan1 = pd.DataFrame({'Prediksi': y_pred, 'Aktual':
y_actual})

print(hasil_perbandingan1)

#hasil_perbandingan1.to_excel('hasil_perbandingan sop.xlsx', index
=False)

# Evaluasi Performa

accuracy = accuracy_score(y_actual, y_pred)

precision = precision_score(y_actual, y_pred, average='macro', zero
_division=0)

recall = recall_score(y_actual, y_pred, average='macro', zero_divis
ion=0)

f1 = f1_score(y_actual, y_pred, average='macro', zero_division=0)

# Tampilkan Hasil Performa

print(f"Accuracy: {accuracy*100:.2f}%")

print(f"Precision: {precision*100:.2f}%")

print(f"Recall: {recall*100:.2f}%")

print(f"F1-score: {f1*100:.2f}%")

# Tampilkan Classification Report yang memberikan detail metrik ev
aluasi per kelas

print("Classification Report:")

print(classification_report(y_actual, y_pred))

cm1=confusion_matrix(y_actual, y_pred)

print("Hasil cm:")

```

```
print(cm1)
```

CONFUSION MATRIX SEBELUM DI OPTIMASI

```
[ ]

cm = confusion_matrix(y_actual, y_pred, labels=model.classes_)

# Mengubah Confusion Matrix menjadi DataFrame untuk kejelasan

cm_df = pd.DataFrame(cm, index=model.classes_, columns=model.class
es_)

# Membuat tampilan visual Confusion Matrix

plt.figure(figsize=(6, 5))

sns.set(font_scale=1.0)

sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues', cbar=False)

plt.xlabel('Prediksi')

plt.ylabel('Aktual')

plt.title('Sebelum Dioptimasi')

plt.show()
```

PENGUJIAN DATA UJI BARU SETELAH DIOPTIMASI

```
[ ]

import pandas as pd

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import accuracy_score, precision_score, recal
l_score, f1_score, classification_report, confusion_matrix

from sklearn.preprocessing import MinMaxScaler

from sklearn.feature_selection import SelectKBest, chi2, f_classif

from sklearn.model_selection import GridSearchCV

# Baca dataset dengan data cuaca yang diketahui (data training)

data1 = pd.read_excel('DATA_CUACA1.xlsx')
```

```

# Menghapus kolom yang tidak diperlukan
data1 = data1.drop(['Tanggal', 'AAT'], axis=1)

# Menghapus baris dengan nilai kosong (NaN)
data1 = data1.dropna()

data1 = data1.drop_duplicates()

data1.info()

# Memisahkan fitur dan target
X_train = data1.drop('Kondisi Cuaca', axis=1)
y_train = data1['Kondisi Cuaca']

# Baca dataset dengan data cuaca yang ingin diprediksi (data testing)
data2 = pd.read_excel('DATA_PRED.xlsx')

data2 = data2.drop(['Tanggal', 'AAT'], axis=1)

X_test2 = data2.drop('Kondisi Cuaca', axis=1)

# Baca dataset dengan data cuaca aktual (data aktual)
data3 = pd.read_excel('DATA_AKTUAL.xlsx')

y_actual2 = data3['Kondisi Cuaca'] # Data cuaca aktual untuk evaluasi

# Normalisasi fitur menggunakan MinMaxScaler
scaler = MinMaxScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test2)

import pandas as pd

df1 = pd.DataFrame(X_test_scaled, columns=X_test.columns)

display(df1)

# Seleksi fitur menggunakan SelectKBest
selector = SelectKBest(score_func=f_classif, k=6)

X_train_selected = selector.fit_transform(X_train_scaled, y_train)
X_test_selected = selector.transform(X_test_scaled)

```

```

# Mendapatkan fitur terbaik yang dipilih
selected_features = X_train.columns[selector.get_support()]

print("Fitur Terbaik:")

print(selected_features)

# Inisialisasi model Naive Bayes
model = GaussianNB()

# Hyperparameter tuning dengan GridSearchCV
param_grid = {
    'var_smoothing': [ 1e-9, 1e-8, 1e-7, 1e-6, 1e-5]
}

grid_search = GridSearchCV(estimator=model, param_grid=param_grid,
    cv=10, scoring='accuracy')

grid_search.fit(X_train_selected, y_train)

# Dapatkan model terbaik setelah hyperparameter tuning
best_model = grid_search.best_estimator_

# Lakukan prediksi pada data cuaca yang ingin diprediksi
y_pred2 = best_model.predict(X_test_selected)

# Tampilkan hasil prediksi
print("Prediksi Cuaca:")

print(y_pred2)

# Tampilkan data cuaca aktual
print("Data Cuaca Aktual:")

print(y_actual2)

# Bandingkan hasil prediksi dengan data cuaca aktual
hasil_perbandingan = pd.DataFrame({'Prediksi': y_pred2, 'Aktual':
y_actual2})

display(hasil_perbandingan)

#hasil_perbandingan.to_excel('hasil_perbandingan op.xlsx', index=False)

# Evaluasi Performa

```

```

accuracy = accuracy_score(y_actual2, y_pred2)

precision = precision_score(y_actual2, y_pred2, average='macro', zero_division=0)

recall = recall_score(y_actual2, y_pred2, average='macro', zero_division=0)

f1 = f1_score(y_actual2, y_pred2, average='macro', zero_division=0)

# Tampilkan Hasil Performa

print(f"Accuracy: {accuracy*100:.2f}%")

print(f"Precision: {precision*100:.2f}%")

print(f"Recall: {recall*100:.2f}%")

print(f"F1-score: {f1*100:.2f}%")

# Tampilkan Classification Report yang memberikan detail metrik evaluasi per kelas

print("Classification Report Setelah Optimasi:")

print(classification_report(y_actual2, y_pred2))

cm1=confusion_matrix(y_actual2, y_pred2)

print("Hasil cm:")

print(cm1)

```

CONFUSION MATRIX SETELAH DIOPTIMASI

```

[]

cm = confusion_matrix(y_actual2, y_pred2, labels=best_model.classes_)

# Mengubah Confusion Matrix menjadi DataFrame untuk kejelasan

cm_df = pd.DataFrame(cm, index=best_model.classes_, columns=best_model.classes_)

# Membuat tampilan visual Confusion Matrix

plt.figure(figsize=(6, 5))

sns.set(font_scale=1.0)

```

```
sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues', cbar=False)

plt.xlabel('Prediksi')

plt.ylabel('Aktual')

plt.title('Sesudah Dioptimasi')

plt.show()
```