

ALGORITMA *K-NEAREST NEIGHBOR* PADA KASUS DATASET *IMBALANCED* UNTUK KLASIFIKASI KINERJA KARYAWAN PERUSAHAAN

Fitri Nuraeni^{*1}, Dede Kurniadi², Moch Haiqal Diazki³

^{1,2,3}Institut Teknologi Garut, Kabupaten Garut
Email: ¹fitri.nuraeni@itg.ac.id, ²dede.kurniadi@itg.ac.id, ³1906057@itg.ac.id
^{*}Penulis Korespondensi

(Naskah masuk: 29 November 2023, diterima untuk diterbitkan: 14 Juni 2024)

Abstrak

Perusahaan perlu menilai kinerja karyawan mereka untuk berbagai tujuan, termasuk promosi jabatan. Namun, data karyawan yang semakin rumit dapat membuat proses penilaian ini menjadi sulit. Penelitian ini bertujuan untuk membuat model *machine learning* yang dapat memprediksi apakah karyawan berpotensi untuk dipromosikan atau tidak. Penelitian ini menggunakan algoritma *K-Nearest Neighbor* dengan menerapkan tahapan-tahapan *Machine Learning LifeCycle (MLLC)*. Untuk mengatasi masalah ketidakseimbangan label kelas dalam *dataset*, teknik *Synthetic Minority Over-sampling Technique (SMOTE)* digunakan. Hasil dari penelitian ini, model dibangun dengan melakukan pemisahan data menggunakan *cross validation* dan menggunakan nilai $k=2$ dalam implementasi algoritma *K-Nearest Neighbor*. Hasil evaluasi model menunjukkan kinerja yang sangat baik dengan nilai akurasi 94%, nilai presisi 90,8%, dan nilai *recall* 97,4%. Model ini juga memiliki kurva *ROC* yang baik yang hampir menyentuh sudut kiri atas dan nilai *AUC* sebesar 0,94 dan nilai *F-Score* sebesar 0,938 yang termasuk ke dalam kategori *excellent*.

Kata kunci: *Imbalanced Dataset*, Kinerja Karyawan, Klasifikasi, *Machine Learning*, *MLLC*.

CLASSIFICATION OF COMPANY EMPLOYEE PERFORMANCE USING *K-NEAREST NEIGHBOR* ALGORITHM AND *SMOTE* METHOD

Abstract

Companies need to assess the performance of their employees for various purposes, including promotions. However, increasingly complex employee data can make this assessment process more accessible. This research aims to create a machine-learning model that can predict whether employees have the potential to be promoted or not. This research uses the *K-Nearest Neighbor* algorithm, with step by step from *Machine Learning Life Cycle (MLLC)* method. To overcome the problem of class label imbalance in the dataset, the *Synthetic Minority Over-sampling Technique (SMOTE)* technique is used. As a result of this research, the model was built by separating the data using *cross-validation* and the value $k=2$ in implementing the *K-Nearest Neighbor* algorithm. The model evaluation results show excellent performance with an accuracy value of 94%, a precision value of 90.8%, and a recall value of 97.4%. In addition, the confusion matrix evaluation showed that only 562 of the 9,377 data-testing did not match the classification results. This model also has a good *ROC* curve, which almost touches the top left corner, and an *AUC* value of 0.94, which is included in the excellent category.

Keywords: *Imbalanced Dataset*, Employee Performance, Classification, *Machine Learning*, *MLLC*.

1. PENDAHULUAN

Penilaian kinerja adalah proses penting bagi perusahaan untuk memahami bakat karyawan, merencanakan pengembangan karir, dan membuat keputusan terkait kompensasi dan promosi (Regina, Sutinah & Agustina, 2021). Penilaian kinerja yang dilakukan dengan benar dan profesional dapat meningkatkan loyalitas dan motivasi karyawan, sehingga memungkinkan tercapainya tujuan perusahaan (Iryani, 2023).

Perusahaan menghadapi tantangan dalam mengevaluasi kinerja karyawan, seperti kompleksitas data karyawan dan sulitnya mengidentifikasi faktor-faktor yang berkontribusi terhadap kinerja karyawan (Regina, Sutinah & Agustina, 2021). Oleh karena itu, diperlukan pendekatan yang efektif untuk mengevaluasi kinerja karyawan, salah satunya adalah dengan menggunakan *machine learning* (Waring, Lindvall & Umeton, 2020). Karena *machine learning* dapat menjadi solusi yang efektif untuk mengatasi tantangan dalam mengevaluasi kinerja karyawan.

Teknik ini dapat membantu perusahaan untuk mengukur kinerja karyawan secara lebih obyektif dan akurat, serta mengidentifikasi karyawan dengan potensi kinerja tinggi.

Berdasarkan permasalahan diatas, ada beberapa penelitian terkait penelitian yang akan dilakukan antara lain, penelitian yang dilakukan oleh (Iryani, 2023) dimana pada penelitian ini menerapkan algoritma *C4.5* untuk membuat model klasifikasi *machine learning* untuk melakukan klasifikasi terhadap data penilaian kinerja karyawan. Hasil dari penelitian ini memperoleh nilai akurasi sebesar 98,51%. Penelitian kedua dilakukan oleh (Siahaan, 2021) dimana pada penelitian ini menerapkan algoritma *Random Forest*, *Decision Tree*, *K-Nearest Neighbor*, *Naïve Bayes*, dan *Logistic Regression* untuk memprediksi penilaian kinerja karyawan. Hasil dari penelitian ini menetapkan bahwa model algoritma *Random Forest* merupakan model yang paling efektif dalam melakukan klasifikasi dan prediksi penilaian kinerja karyawan kontrak yang memiliki nilai akurasi sebesar 90,62%, presisi 72,22%, *recall* 75% dan nilai *F1 score* 71,11%. Penelitian ketiga dilakukan oleh (Anggara, Widjaja & Suteja, 2022) dimana pada penelitian ini menerapkan algoritma *Logistic Regression* dan *Decision Tree* untuk membuat model klasifikasi untuk memprediksi kinerja karyawan sebagai rekomendasi kenaikan jabatan karyawan di Yayasan Prasama Bhakti. Hasil dari penelitian ini memperoleh nilai akurasi sebesar 90%. Penelitian keempat dilakukan oleh (Sotarjua & Santoso, 2022) dimana pada penelitian ini menerapkan algoritma *K-Nearest Neighbor*, *Decision Tree*, *Random Forest* untuk mengevaluasi seberapa baik model *machine learning* bekerja dengan data klasifikasi promosi karyawan. Hasil dari penelitian ini menetapkan bahwa model algoritma *K-Nearest Neighbors* menjadi yang terbaik untuk digunakan dalam mengklasifikasi promosi karyawan dengan nilai akurasi sebesar 88,51%, presisi 92,11% dan *recall* 84,25%. Penelitian kelima dilakukan oleh (Wirayasa & Santoso, 2022) dimana pada penelitian ini menerapkan algoritma *K-Means* dan algoritma *Classifier* untuk mengevaluasi seberapa baik kinerja kedua algoritma ini ketika membuat *cluster* kelas yang sesuai dan menerapkan metode *Principal Component Analysis (PCA)* untuk mengurangi jumlah variabel dalam *set data*. Hasil dari penelitian ini diperoleh nilai akurasi sebesar 98% dan *AUC* rata-rata 0,99. Penelitian keenam dilakukan oleh (Laga, 2023) dimana pada penelitian ini menerapkan algoritma *K-Nearest Neighbors* dan *Support Vector Machine* untuk melakukan klasifikasi kinerja karyawan. Hasil dari penelitian ini menetapkan model algoritma *K-Nearest Neighbors* terbukti menjadi yang paling efektif dalam klasifikasi kinerja karyawan dalam penelitian ini dengan akurasi sebesar 90,13%, presisi 91% dan *recall* 98,95%.

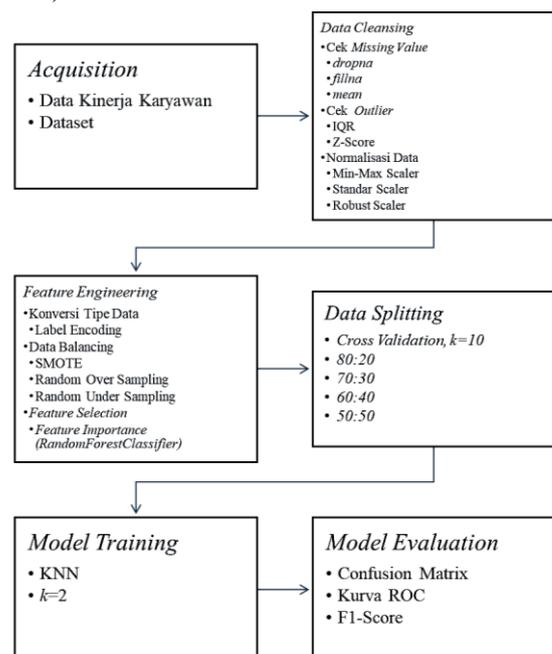
Mengacu pada penelitian rujukan diatas, tujuan dari penelitian ini yaitu untuk membangun model

machine learning dalam mengklasifikasi dan memprediksi kinerja karyawan berdasarkan data penilaian kinerja karyawan menggunakan algoritma *K-Nearest Neighbor*, karena pada penelitian yang dilakukan (Laga, 2023) menunjukkan model yang dibuat menggunakan algoritma *K-Nearest Neighbor* menghasilkan akurasi yang tinggi sebesar 90,13%.

Dataset yang digunakan pada penelitian ini diperoleh dari *website* yang menyediakan kumpulan *dataset public* yaitu *website Kaggle* dengan nama *HR Analytics: Employee Promotion Data* (Mobius, 2020). *Dataset* ini memiliki karakteristik *imbalanced class* pada label kelas *dataset*, sehingga diperlukan penyeimbangan data menggunakan metode *SMOTE* yang terbukti dapat mengatasi ketidakseimbangan data serta dapat meningkatkan kinerja model klasifikasi (Kumalasari & Merdekawati, 2023). Hasil dari penelitian ini berupa model *machine learning* yang nantinya diharapkan dapat membantu perusahaan dalam mengoptimalkan evaluasi kinerja karyawan dan mengidentifikasi karyawan yang kemungkinan memiliki kinerja baik atau buruk, serta memberikan rekomendasi dalam pengambilan keputusan terkait promosi karyawan.

2. METODE PENELITIAN

Penelitian ini menggunakan metode *Machine Learning LifeCycle (MLLC)* adalah prosedur untuk membuat model *machine learning*, yang dimulai dengan pengumpulan data dan berakhir ketika model tersebut siap digunakan, sebagaimana disajikan pada Gambar 1. Sifat *machine learning lifecycle* yang berulang dan berwawasan ke depan berasal dari tujuan setiap iterasi untuk meningkatkan kinerja dan akurasi model dari waktu ke waktu (Ibnu Daqiqil Id, 2021).



Gambar 1. Kerangka Penelitian

Tiga tahap ditunjukkan pada Gambar 1 meliputi input, proses, dan output. Sumber dari mana output dihasilkan disebut input. Proses mengubah informasi menjadi output dikenal sebagai proses. Di sisi lain, output adalah hasil dari pemrosesan input. Pada tahapan *process* terdiri dari 6 tahapan metode *MLLC* yaitu *Data Acquisition*, *Data Cleansing*, *Feature Engineering*, *Data Split*, *Model Training*, *Model Evaluation*.

2.1. *K-Nearest Neighbor*

Di antara metode yang digunakan dalam masalah klasifikasi adalah *K-Nearest Neighbor* (*KNN*). Menemukan jarak terpendek antara data yang akan dievaluasi dan tetangga terdekat dalam data pelatihan adalah dasar dari algoritma *KNN* (Nikmatun & Waspada, 2019). Berikut ini adalah rumus umum untuk menentukan jarak dalam hal jarak *Euclidean*:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

Keterangan:

D = jarak antara x dan y

i = indeks dari atribut

n = jumlah atribut

x_i = atribut ke- i pada data x , ($i=1, 2, 3, \dots, n$)

y_i = atribut ke- i pada data y , ($i=1, 2, 3, \dots, n$)

Proses pemahaman algoritma *K-Nearest Neighbor* memiliki beberapa tahapan yaitu:

1. Tetapkan nilai K . Untuk mencegah hasil pemungutan suara yang seimbang, K harus memiliki nilai bilangan asli ganjil.
2. Mencari tahu seberapa jauh titik baru tersebut dari semua data pelatihan.
3. Pilih K titik yang paling dekat dengan titik baru.
4. Ketika mengklasifikasikan data, titik terdekat ditentukan berdasarkan kategori atau kelas. Kategori dengan jumlah titik terdekat yang paling banyak akan menjadi titik baru. Titik baru akan berada pada nilai rata-rata dari K tetangga terdekat dalam situasi regresi.

2.2. *Synthetic Minority Over-sampling Technique (SMOTE)*

Untuk mengurangi ketidakseimbangan kelas, Teknik *Synthetic Minority Over-sampling Technique* (*SMOTE*) sering digunakan dengan mengambil sampel ulang dari kelas minoritas, strategi ini mengumpulkan sampel tambahan dari kelompok yang kurang terwakili untuk menyeimbangkan kumpulan data (Siringoringo, 2018). Metode ini dapat digunakan untuk menambahkan data kelas minoritas dengan menciptakan data sintetis yang tidak ada di dataset asli (Nursyahfitri, Rozikin and Adam, 2022).

Cara kerja *SMOTE* melalui beberapa tahapan (A. Rahim, Ingrid Yanuar Risca Pratiwi and Muhammad Ainul Fikri, 2023): a) mengidentifikasi kelas minoritas; b) mencari kelompok data yang

berdekatan dengan kelas minoritas; c) menciptakan data sintetis: dari kelompok data yang berdekatan dengan kelas minoritas, yang dapat menambahkan data kelas minoritas dalam dataset imbalanced; d) mengubah kelas dari data sintetis yang dibuat menjadi kelas minoritas; dan e) mengulangi proses oversampling untuk menciptakan lebih banyak data sintetis.

2.3. Uji Performa

Uji performa menggunakan *confusion matrix* untuk melakukan perhitungan performa model klasifikasi. Informasi tentang kategori yang diprediksi dengan benar oleh sistem klasifikasi terdapat dalam Tabel 1 *confusion matrix* (Gunawan, Pratiwi & Pratama, 2018). Pengujian ini dapat menghasilkan performa model berupa nilai akurasi, presisi dan *recall* dapat dicari menggunakan persamaan berikut:

Tabel 1. *Confusion Matrix*

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FN
	Negative	FP	TN

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

Keterangan:

TP = *True Positive*

FP = *False Positive*

TN = *True Negative*

FN = *False Negative*

Kemudian melakukan pengujian performa dengan menggunakan kurva *ROC* dan nilai *AUC* untuk mengatur, menampilkan dan memilih pengklasifikasi sesuai dengan kinerja algoritma (Arifin & Syalwah, 2020). Nilai *Area Under Curve* (*AUC*) dapat dibagi menjadi beberapa kelompok yaitu:

1. 0.90 – 1.00 = *Excellent Classification*
2. 0.80 – 0.90 = *Good Classification*
3. 0.70 – 0.80 = *Fair Classification*
4. 0.60 – 0.70 = *Poor Classification*
5. 0.50 – 0.60 = *Failure*

2.4. Sumber Data

Dataset yang digunakan pada penelitian ini yaitu *dataset public* karena data tersebut dapat digunakan untuk mewakili secara luas dari kriteria penilaian kinerja karyawan dari suatu perusahaan. *dataset* didapatkan dari *website* yang menyediakan kumpulan *dataset public* yaitu *website Kaggle* dengan nama *HR Analytics: Employee Promotion Data* (Mobius, 2020). *Dataset* ini memiliki 54808

records atau data dan 13 *feature* atau atribut antara lain *employee_id*, *department*, *region*, *education*, *gender*, *recruitment_channel*, *no_of_trainings*, *age*, *previous_year_rating*, *length_of_service*, *awards_won?*, *avg_training_score* dan *is_promoted*. Sampel dari *dataset* disajikan pada Tabel 2 dibawah ini.

Tabel 2. Sampel Data HR Analytics: Employee Promotion Data

No.	EID	...	LOS	AW	ATS	IP
1.	8724	...	1	0	77	1
2.	74430	...	5	0	51	0
3.	72255	...	4	0	47	0
4.	38562	...	9	0	65	0
5.	64486	...	7	0	61	0
...
54808	5973	...	5	0	89	1

Keterangan:

EID = *employee_id*

LOS = *length_of_service*

AW = *awards_won?*

ATS = *avg_training_score*

IP = *is_promoted*

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan dari penelitian ini menggunakan metode SMOTE dan algoritma *K-Nearest Neighbor*. *Google Colab* merupakan salah satu *tools* tambahan yang digunakan untuk membuat model (Guntara, 2023). Sub-bahasan berikut ini menyajikan hasil penelitian dan pembahasannya sebagai berikut:

3.1. Hasil Penelitian

Dalam penelitian ini, metode pendekatan *MLLC* diterapkan. Hasil dari penelitian ini akan dijelaskan dengan menggunakan tahapan-tahapan *MLLC*. Untuk lebih jelasnya dapat dilihat pada sub poin berikut:

3.1.1. Data Acquisition

Proses memahami data merupakan langkah awal dalam penelitian ini, dan didasarkan pada objek penelitian yang dipilih. Pengumpulan data dan identifikasi data adalah aktivitas penelitian yang dilakukan.

Dataset yang digunakan pada penelitian ini memiliki 54.808 *records* dan 13 atribut, dengan penjelasan dari setiap atribut:

- employee_id*, menjelaskan *id* unik dari karyawan sehingga tidak akan terjadi kesamaan antara karyawan satu dengan yang lainnya.
- department*, menjelaskan dari departemen mana karyawan bekerja.
- region*, menjelaskan wilayah kerja karyawan.
- education*, menjelaskan pendidikan terakhir dari karyawan.
- gender*, menjelaskan jenis kelamin dari karyawan.

- Recruitment_channel*, menjelaskan darimana karyawan tersebut mendapatkan informasi rekrutmen karyawan.
- no_of_trainings*, menjelaskan jumlah pelatihan yang telah dilakukan oleh karyawan.
- age*, menjelaskan umur dari karyawan.
- previous_year_rating*, menjelaskan peringkat dari karyawan pada tahun sebelumnya.
- length_of_service*, menjelaskan masa jabatan karyawan.
- awards_won?*, menjelaskan apakah karyawan mendapatkan penghargaan dari perusahaan pada tahun sebelumnya atau tidak.
- avg_training_score*, menjelaskan skor rata-rata dalam evaluasi penilaian karyawan.
- is_promoted*, menjelaskan apakah karyawan tersebut mendapatkan promosi atau tidak. Atribut ini merupakan label kelas pada model yang dibuat.

3.1.2. Data Cleansing

Pada tahapan ini, data yang sudah didapatkan akan dibersihkan atau disesuaikan agar kualitas data meningkat dan data dapat digunakan. Proses ini dilakukan untuk menghindari hasil klasifikasi yang tidak sesuai, seperti *output* yang tidak sesuai, model yang tidak akurat, dan lain sebagainya (Susana et al., 2022).

Pada tahapan ini, dilakukan uji coba berkali-kali untuk menentukan teknik terbaik yang diterapkan pada model klasifikasi KNN menggunakan nilai $k=2$. Berikut merupakan beberapa proses yang harus dilakukan dalam *data cleansing* pada penelitian ini sebagai berikut:

- Cek *Missing Values*

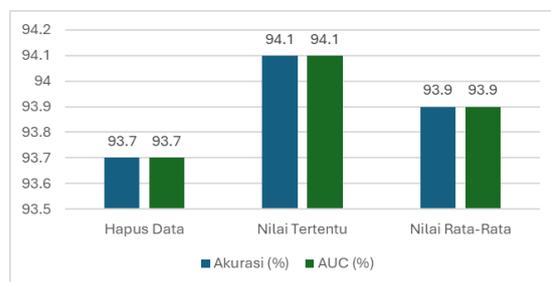
Pada proses ini, dilakukan identifikasi kolom atau baris yang memiliki nilai kosong atau hilang. Kondisi ini muncul ketika ada kekurangan informasi, kesulitan menemukan informasi, atau tidak adanya informasi mengenai objek tersebut (Prayoga, Nawangsih & Wiyatno, 2019).

Tabel 3. Hasil Identifikasi *Missing Values*

No.	Atribut	Missing Values
1.	<i>employee_id</i>	0
2.	<i>department</i>	0
3.	<i>region</i>	0
4.	<i>education</i>	0
5.	<i>gender</i>	0
6.	<i>recruitment_channel</i>	0
7.	<i>no_of_trainings</i>	0
8.	<i>age</i>	0
9.	<i>previous_year_rating</i>	4124
10.	<i>length_of_service</i>	0
11.	<i>awards_won?</i>	0
12.	<i>avg_training_score</i>	0
13.	<i>is_promoted</i>	0

Tabel 3 diatas menjelaskan bahwa data yang digunakan memiliki *missing values* pada atribut *previous_year_rating* sehingga perlu dibersihkan. Selanjutnya akan dilakukan percobaan untuk mencari teknik pembersihan *missing values* yang paling baik,

agar performa model menjadi lebih baik dan mengurangi kompleksitas pada model. Pada proses ini dipilih 3 jenis metode penanganan *missing values*, yaitu: a) *dropna*, metode yang digunakan untuk mengeluarkan data yang hilang dari dataset, metode ini mengurangi ketergantungan pada metode penggantian data namun dapat mengurangi ketepatan model (Nursyahfitri, Rozikin and Adam, 2022); b) *fillna*, metode ini mengisi data yang hilang dengan nilai tertentu, namun metode ini hanya sesuai untuk kasus tertentu (Nugraha et al., 2023); dan c) *mean*, metode ini mengisi data yang hilang dengan rata-rata nilai data yang tidak hilang, namun hanya bisa digunakan pada data numerik saja (Yulian Pamuji et al., 2024).



Gambar 2. Hasil Percobaan Teknik Pembersihan *Missing Values*

Gambar 2 menjelaskan sumbu y merupakan nilai akurasi dan AUC, sedangkan sumbu x merupakan teknik pembersihan *missing values* yang digunakan. Dapat dilihat bahwa teknik pembersihan *missing values* dengan mengisi data yang hilang dengan nilai tertentu mendapatkan nilai akurasi dan AUC yang baik, sehingga penelitian ini akan menggunakan teknik pembersihan *missing values* dengan mengisi data yang hilang dengan nilai tertentu.

b. Cek *Outlier*

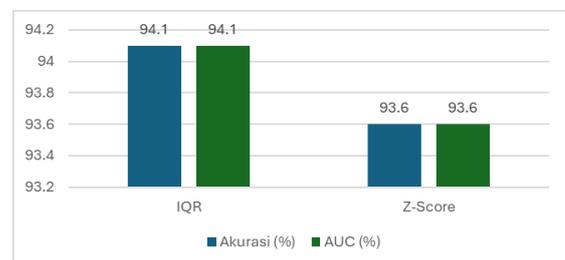
Pada proses ini, dilakukan identifikasi terhadap *dataset* apakah terdapat *outlier* atau tidak. Adanya data *outlier* ini dapat memberi efek bagi hasil akhir dari proses klasifikasi (Sihombing et al., 2023).



Gambar 3. Hasil Identifikasi *Outlier*

Gambar 3 menjelaskan sumbu y merupakan nilai dari atribut dan sumbu x merupakan atributnya.

Dapat dilihat bahwa sebaran nilai pada atribut *length_of_service* terdapat data yang terpisah dari *box*, sehingga perlu dilakukan penanganan *outlier*. Selanjutnya akan dilakukan percobaan untuk mencari metode penanganan *outlier* yang paling baik, agar performa model menjadi lebih baik dan mengurangi kompleksitas pada model. Metode yang dapat digunakan untuk menangani *outlier* pada dataset, yaitu: a) *Interquartile Range (IQR)*, dimana *outlier* dapat disimpulkan jika nilai data lebih kecil dari tahap ketiga atau lebih besar dari tahap keempat; dan b) *Z-Score*, dimana *outlier* dapat disimpulkan jika nilai data lebih kecil dari nilai rata-rata minus standar deviasi atau lebih besar dari nilai rata-rata plus standar deviasi (Dastjerdy, Saeidi and Heidarzadeh, 2023).

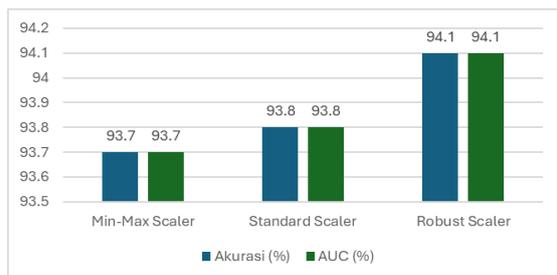


Gambar 4. Hasil Percobaan Metode Penanganan *Outlier*

Gambar 4 menjelaskan sumbu y merupakan nilai akurasi dan AUC sedangkan sumbu x merupakan metode penanganan *outlier* yang digunakan. Dapat dilihat bahwa metode *IQR* mendapatkan nilai akurasi dan AUC yang baik, sehingga penelitian ini akan menggunakan metode *IQR* untuk menangani *outlier*.

c. Normalisasi Data

Pada proses ini dilakukan normalisasi data untuk mengubah nilai numerik dari sebuah atribut menjadi sebuah nilai yang lebih umum agar model yang dihasilkan lebih stabil. Dalam proses normalisasi ini akan dilakukan percobaan terlebih dahulu untuk mencari metode normalisasi yang paling baik, agar performa model menjadi lebih baik dan mengurangi kompleksitas pada model. Teknik normalisasi yang umum digunakan yaitu: a) *Min-Max Normalization*, metode normalisasi yang merubah rentang nilai data menjadi antara 0 dan 1 (Permana and Salisah, 2022); b) *Standard Scaler*, metode ini mengubah nilai data ke dalam skala standar deviasi, yang merupakan nilai yang menunjukkan jarak data dari rata-rata data (Sumantri, Novianto and Prihastuti, 2023); dan c) *robust scaler*, metode normalisasi data yang dapat mengurangi efek ketidakhomogenitas yang dapat berpengaruh pada kinerja model klasifikasi (Virantika, Kusnawi and Ipmawati, 2022).



Gambar 5. Hasil Percobaan Metode Normalisasi

Gambar 5 menjelaskan sumbu y merupakan nilai akurasi dan *AUC* dan sumbu x merupakan metode normalisasi yang digunakan. Dapat dilihat bahwa metode normalisasi dengan metode *Robust Scaler* mendapatkan nilai akurasi dan *AUC* yang baik, sehingga penelitian ini akan menggunakan metode *Robust Scaler* untuk melakukan normalisasi data.

3.1.3. Feature Engineering

Pada tahapan ini, data yang sudah melalui tahapan *data cleansing* akan dilakukan penyeimbangan data untuk mengatasi adanya ketidakseimbangan jumlah kelas pada *dataset*. Pada tahapan ini, dilakukan uji coba berkali-kali untuk menentukan teknik terbaik yang diterapkan pada model klasifikasi KNN menggunakan nilai $k=2$. Berikut merupakan beberapa proses yang dilakukan dalam *feature engineering* pada penelitian ini sebagai berikut:

a. Konversi Tipe Data

Pada proses ini, dilakukan konversi pada tipe data atribut. Transformasi data dilakukan data berguna untuk model pembelajaran mesin, yang terbatas pada pemrosesan data non-numerik. Berdasarkan penelitian (Rashed-Al-Mahfuz et al., 2021), data non-numeric dikonversi menjadi data numerik, dengan teknik *Label Encoding*. Teknik ini dapat digunakan dalam pengolahan data yang bersifat non-ordinal, dimana kelas tidak tergantung urutan (Herdian, Kamila and Agung Musa Budidarma, 2024). Metode ini dipilih karena mampu mengubah data kategorikal tanpa menambahkan variabel baru atau meningkatkan dimensi kumpulan data, yang dapat bermanfaat untuk efisiensi komputasi (Hancock and Khoshgoftaar, 2020). Contoh penggunaan Teknik *label encoding*, dapat terlihat pada tabel 4 dan tabel 5.

Tabel 4. Perbandingan Atribut *departement*

No.	Nilai Teks	Nilai Numerik
1.	<i>Analytics</i>	0
2.	<i>Finance</i>	1
3.	<i>HR</i>	2
4.	<i>Legal</i>	3
5.	<i>Operations</i>	4
6.	<i>Procurement</i>	5
7.	<i>R&D</i>	6
8.	<i>Sales & Marketing</i>	7
9.	<i>Technology</i>	8

Proses konversi yang dilakukan karena model yang dibuat hanya dapat menerima data dalam bentuk numerik. Tabel 4 diatas menjelaskan bahwa data yang sebelumnya berbentuk teks berubah menjadi numerik dengan nilai 0 sampai 8 contohnya seperti *Analytics* berubah menjadi 0.

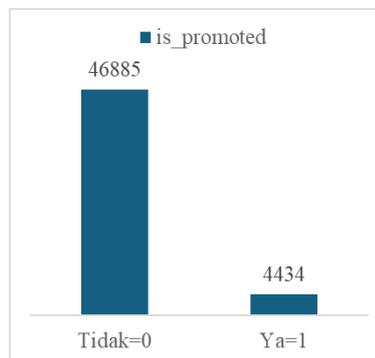
Tabel 5. Perbandingan Atribut *gender*

No.	Nilai Teks	Nilai Numerik
1.	<i>f (Female)</i>	0
2.	<i>m (Male)</i>	1

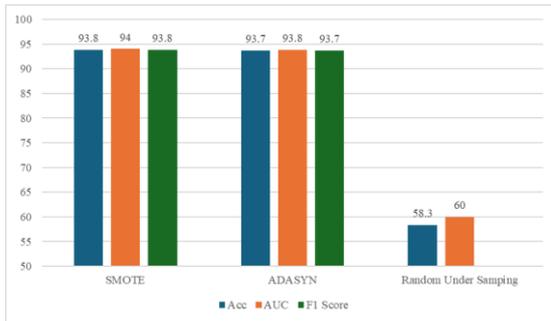
Tabel 7 diatas menjelaskan bahwa data yang sebelumnya berbentuk teks diubah menjadi numerik dengan nilai 0 sampai 1 seperti *f (Female)* berubah menjadi 0 dan *m (Male)* berubah menjadi 1.

b. Data Balancing

Pada proses ini, dilakukan penyeimbangan data karena terdapat *imbalanced class* pada label *dataset*. Proses ini dilakukan dengan tujuan agar hasil klasifikasi tidak condong ke kelas mayoritas (Mutmainah, 2021). Proses resampling digunakan untuk menghasilkan jumlah antara kelas 0 dan kelas 1 yang sama yaitu 46.885 data, dari yang sebelumnya berjumlah 46.885 untuk kelas 0 dan 4.434 untuk kelas 1 sesuai yang terlihat pada gambar 6.

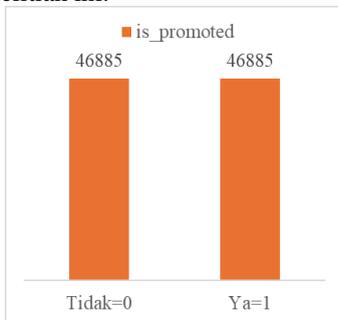
Gambar 6. Perbandingan jumlah data sebelum dilakukan *Balancing*

Selanjutnya, dilakukan percobaan untuk membandingkan metode *balancing* yang lebih baik untuk digunakan. Metode yang dibandingkan yaitu (Nurhopipah and Magnolia, 2023): a) SMOTE adalah teknik *oversampling* yang digunakan untuk meningkatkan jumlah sampel pada kelas minoritas dengan menginterpolasi atribut-atribut dari sampel-sampel yang ada; b) *Adaptive Synthetic Sampling* (ADASYN) menciptakan lebih banyak sampel sintetis di area yang kurang berpenghuni; dan *Random Under Sampling*, yaitu secara acak mengurangi jumlah sampel dari kelas mayoritas hingga jumlahnya sebanding dengan jumlah sampel kelas minoritas. Berikut merupakan hasil percobaan untuk membandingkan metode *balancing* yang paling baik disajikan pada Gambar 7. berikut ini:



Gambar 7. Hasil Percobaan Metode *Balancing*

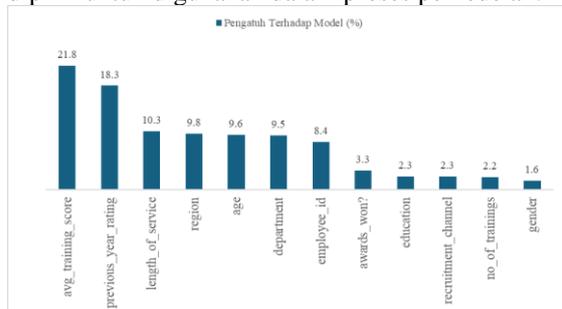
Gambar 7 menjelaskan sumbu y merupakan nilai akurasi dan *AUC* dan sumbu x merupakan teknik *resampling* yang digunakan. Karena terbukti bahwa metode penyeimbangan *SMOTE* menghasilkan nilai akurasi dan *AUC* yang baik, maka *SMOTE* akan digunakan dalam prosedur penyeimbangan data dalam penelitian ini.



Gambar 8. Perbandingan jumlah data setelah dilakukan *Balancing*

c. *Feature Selection*

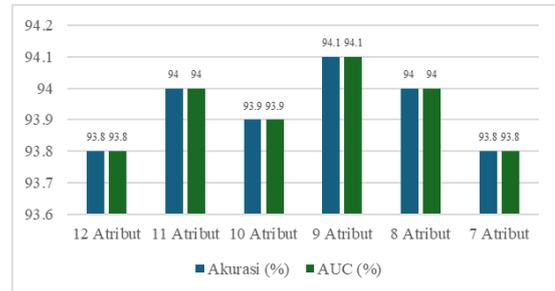
Pada proses ini, dilakukan pemilihan untuk menentukan atribut mana yang paling berpengaruh pada pembuatan model. Proses ini dilakukan dengan tujuan agar memberikan kontribusi yang baik terhadap proses klasifikasi (Karo, Amalia & Septiana, 2022). Metode seleksi fitur yang digunakan adalah metode *feature importance* dengan model *Random Forest Classifier*. Atribut dengan bobot nilai tertinggi dipilih untuk digunakan dalam proses pemodelan.



Gambar 9. Hasil Penerapan *Feature Importance*

Gambar 9 menjelaskan sumbu y merupakan atribut pada data yang digunakan dan sumbu x merupakan nilai pada setiap atribut yang berpengaruh terhadap model. Dapat dilihat bahwa atribut *avg_training_score* merupakan atribut yang memiliki pengaruh paling besar. Selanjutnya, dilakukan percobaan untuk mencari jumlah atribut yang paling

baik, agar performa model menjadi lebih baik dan mengurangi kompleksitas pada model.

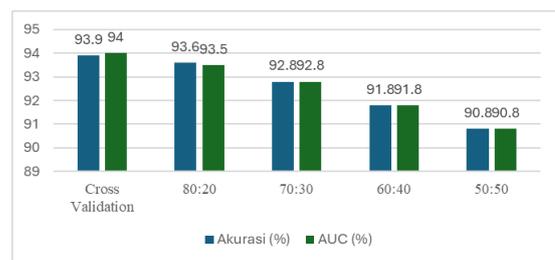


Gambar 10. Hasil Percobaan Dengan Jumlah Atribut yang Digunakan

Gambar 10 menjelaskan sumbu y merupakan nilai akurasi dan nilai *AUC* dan sumbu x merupakan jumlah atribut yang digunakan. Dari keenam percobaan diatas, percobaan dengan 9 atribut memiliki nilai akurasi dan *AUC* yang tinggi, maka pada proses *feature selection* ini terdapat tiga atribut yang harus dihapus yaitu atribut *gender*, *no_of_trainings* dan *recruitment_channel*.

3.1.4. *Data Split*

Pada proses ini, data yang sudah melalui tahapan *data cleansing* dan *feature engineering* akan dilakukan pembagian data menjadi *data training* dan *data testing*. Untuk menentukan rasio dari pembagian data, maka dilakukan percobaan melatih model dengan *cross validation* dengan nilai *k*=10 dan beberapa rasio yaitu 80:20, 70:30, 60:40, 50:50. Performa dari masing-masing model akan dibandingkan dan model dengan performa terbaik maka rasio tersebut akan digunakan pada penelitian ini.



Gambar 11. Hasil Percobaan dari *Split Data*

Gambar 11 menjelaskan sumbu y merupakan nilai akurasi dan nilai *AUC* dan sumbu x merupakan rasio pembagian data. Dapat dilihat hasil yang diperoleh dari percobaan *cross validation* mendapatkan hasil percobaan paling baik, sehingga penelitian ini akan membagi data dengan *cross validation*.

3.1.5. *Model Training*

Pada proses ini, *data training* akan digunakan untuk membuat model untuk menentukan nilai *k* terbaik pada pembuatan model. *K* pada metode *K-*

Nearest Neighbors (KNN) merupakan parameter yang digunakan untuk menentukan jumlah data yang digunakan sebagai referensi untuk mencari jarak terdekat antara data yang akan dievaluasi dan data pelatihan (Andie and Hasanuddin, 2023). Dalam proses mencari nilai k terbaik dilakukan beberapa percobaan untuk nilai k=1 hingga k=10. Nilai k yang memiliki performa paling baik akan digunakan pada penelitian ini. Berikut merupakan hasil percobaan untuk menentukan nilai k yang paling baik disajikan dalam Tabel 6.

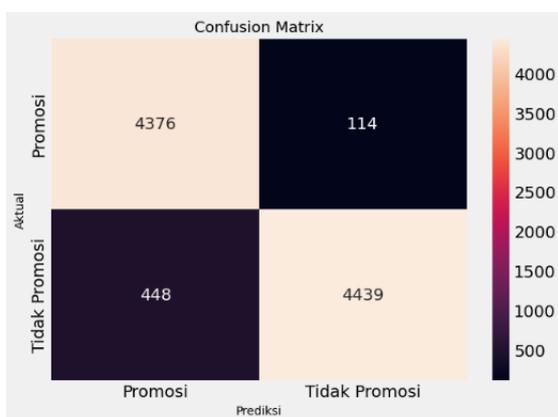
Tabel 6. Hasil Percobaan Dengan Beberapa Nilai K

Nilai K	Nilai Akurasi	Nilai AUC
1	93,6%	93,7%
2	94%	94,1%
3	91,1%	91,2%
4	91,7%	91,9%
5	88,9%	89,2%
6	89,5%	89,7%
7	87,3%	87,6%
8	88,1%	88,4%
9	85,9%	86,2%
10	86,8%	87%

Tabel 6 ini menunjukkan hasil dari percobaan untuk menentukan nilai k yang paling baik. Dapat dilihat hasil yang diperoleh dari percobaan dengan nilai k=2 mendapatkan hasil percobaan paling baik, sehingga penelitian ini akan menggunakan nilai k=2 untuk mendapatkan model yang lebih baik.

3.1.6. Model Evaluation

Pada proses ini, *data testing* dimasukkan kedalam model akhir yang sudah dilatih pada tahapan *model training* untuk dilakukan proses evaluasi. Metode evaluasi yang digunakan yaitu *confusion matrix*, akurasi, presisi, *recall* dan kurva *ROC/AUC*. Untuk mendapatkan nilai akurasi, presisi dan *recall*, model dilakukan evaluasi terlebih dahulu menggunakan metode *confusion matrix*.

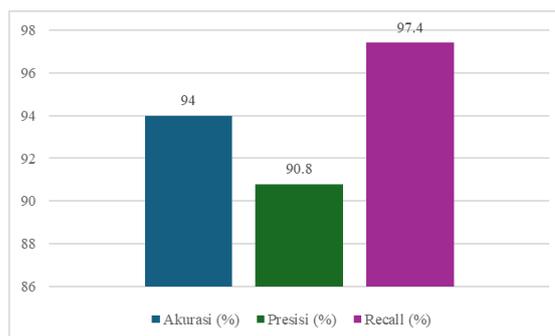


Gambar 7. Hasil Evaluasi *Confusion Matrix*

Gambar 12 menunjukkan bahwa model yang dibuat memiliki performa yang sangat bagus dalam proses klasifikasi. Terdapat 562 data yang mengalami kesalahan dalam proses klasifikasi yang seharusnya data tersebut masuk ke dalam kelas 0 atau promosi,

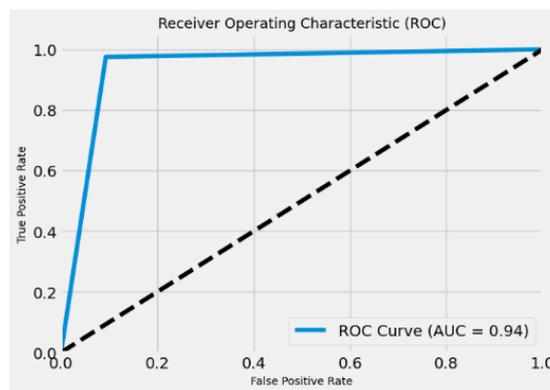
tetapi model mengklasifikasi sebagai kelas 1 atau tidak promosi dan sisanya yaitu 8.815 data dapat diklasifikasikan dengan sangat baik.

Setelah mendapatkan nilai TP, TN, FP dan FN yang didapatkan dari *confusion matrix*, maka selanjutnya mendapatkan nilai akurasi, presisi dan *recall* dimana persamaan (1), persamaan (2), dan persamaan (3) dapat digunakan untuk menghitung nilai akurasi, nilai presisi, dan nilai *recall*. Berikut merupakan nilai akurasi, presisi dan *recall* yang sudah dihitung secara otomatis disajikan dalam Gambar 11. sebagai berikut:



Gambar 8. Nilai Akurasi, Presisi dan *Recall*

Gambar 13 menunjukkan sumbu y merupakan nilai akurasi, presisi dan *recall* dan sumbu x merupakan nilai metrik evaluasinya. Dapat dilihat bahwa model yang telah dibuat berdasarkan nilai k=2 memiliki performa yang bagus dengan nilai akurasi, presisi dan *recall* yang hampir mendekati sempurna. Selain itu, untuk melihat seberapa bagus model yang dibuat dapat membedakan antara kelas satu dengan yang lain dilakukan evaluasi menggunakan kurva *ROC/AUC*.



Gambar 9. Kurva *ROC/AUC*

Gambar 14 menjelaskan sumbu y merupakan nilai TPR dimana proporsi yang benar dikenali pada semua kelas positif yang sebenarnya dan sumbu x merupakan FPR dimana proporsi negatif yang salah dikenali pada semua kelas negatif yang sebenarnya. Garis biru merupakan performa model pada berbagai ambang batas dan garis hitam putus-putus merupakan performa model acak. Dapat disimpulkan bahwa model yang dibuat berdasarkan nilai k=2 dapat

membedakan antar kelas, dengan kurva *ROC* yaitu garis yang berwarna biru berada pada sudut kiri atas dan nilai *AUC* sebesar 0,94 yang mana model yang dibuat termasuk ke dalam kategori *excellent*.

Selanjutnya untuk mendapatkan nilai kinerja model yang lebih akurat, dilakukan evaluasi menggunakan *F1-Score* dengan persamaan (5), mengingat dataset yang digunakan merupakan dataset *imbalanced* dengan menggunakan metode *SMOTE* (Nurhopipah and Magnolia, 2023). Untuk model klasifikasi akhir yang dibuat, didapatkan nilai *F-Score* sebesar 0,938 atau 93,8%. Nilai *F-Measure* akan lebih tinggi jika model klasifikasi lebih baik dalam mengelompokkan data ke kelas yang tepat (Halim and Azmi, 2023).

3.2. Pembahasan Hasil

Hasil dari penelitian ini, model yang dibangun dalam penelitian ini dapat melakukan klasifikasi kinerja karyawan dengan sangat baik. Model ini memiliki akurasi sebesar 94%, dengan hanya 562 data yang mengalami kesalahan dalam proses klasifikasi. Hasil ini juga selaras dengan penelitian sebelumnya, bahwa penerapan metode *SMOTE* dapat mempengaruhi terhadap hasil performa model yang lebih baik (Kurniadi, Nuraeni & Firmansyah, 2022).

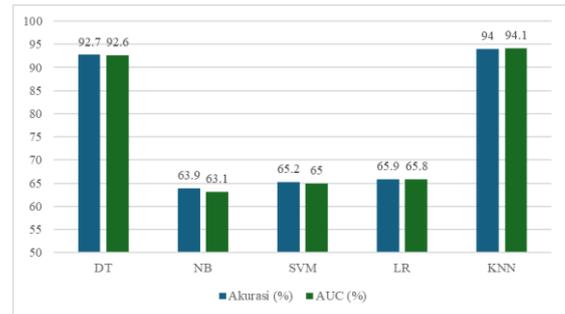
Telah dibuktikan juga pada tahapan *model training*, performa yang dihasilkan pada penelitian ini lebih baik dibandingkan dengan penelitian-penelitian sebelumnya yang menggunakan algoritma dan *dataset* yang berbeda (Siahaan, 2021; Anggara, Widjaja & Suteja, 2022; Sotarjua & Santoso, 2022; Laga, 2023).

Tabel 7. Perbandingan Algoritma Penelitian Sebelumnya

No.	Penelitian	Algoritma	Akurasi
1.	(Siahaan, 2021)	<i>Naive Bayes</i>	90,6%
2.	(Anggara, Widjaja & Suteja, 2022)	<i>Logistic Regression</i>	90%
3.	(Sotarjua & Santoso, 2022)	<i>Decision Tree</i>	88,5%
4.	(Laga, 2023)	<i>SVM</i>	90,1%
5.	Penelitian ini	<i>KNN</i>	94%

Tabel 7 menjelaskan bahwa algoritma *K-Nearest Neighbor* dan *SMOTE* merupakan algoritma yang terbaik. Sebuah percobaan dilakukan dengan menggunakan *dataset* dan prosedur penyesuaian yang sama dengan penelitian ini untuk menentukan apakah benar algoritma yang digunakan dalam penelitian ini adalah yang terbaik jika dibandingkan dengan algoritma lainnya.

Gambar 15 menjelaskan sumbu y merupakan nilai akurasi dan nilai *AUC* dari setiap algoritma dan sumbu x merupakan algoritma pada penelitian sebelumnya dan penelitian ini. Dari beberapa percobaan tersebut, dapat dikatakan bahwa algoritma *K-Nearest Neighbor* merupakan algoritma yang paling baik diantara algoritma-algoritma tersebut.



Gambar 10. Perbandingan Algoritma Dengan *Dataset* yang Sama

4. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat dikatakan bahwa klasifikasi kinerja karyawan perusahaan menggunakan algoritma *K-Nearest Neighbor* dan metode *SMOTE* pada kasus *dataset imbalanced*, dari *dataset* yang didapatkan terdapat beberapa penyesuaian, yaitu melakukan konversi tipe data dengan *label encoder*, mengatasi *missing values* dengan mengisi data yang hilang dengan nilai tertentu, mengatasi *outlier* dengan metode *IQR*, melakukan normalisasi data dengan *Robust Scaler*, melakukan *balancing* pada *imbalanced class* dengan *SMOTE* dan menggunakan 9 atribut dengan tidak menggunakan atribut *gender*, *no_of_trainings* dan *recruitment_channel*, melakukan pembagian data menjadi *data training* dan *data testing* menggunakan *cross validation* dan menggunakan nilai $k=2$ untuk membangun model klasifikasi. Lalu untuk hasil evaluasinya, model yang dibangun memiliki performa yang sangat bagus dan hampir mendekati sempurna, dengan nilai akurasi sebesar 94%, presisi sebesar 90,8%, *recall* sebesar 97,4%, serta hasil evaluasi menggunakan *confusion matrix* hanya ada 562 data yang tidak sesuai hasil klasifikasinya, kurva *ROC* yang bagus yaitu hampir menyentuh sudut kiri atas dan nilai *AUC* sebesar 0,94 yang termasuk ke dalam kategori *excellent*. Selain itu, model ini memiliki nilai *F-Score* sebesar 0,938.

DAFTAR PUSTAKA

- A. RAHIM, A.M., INGGRID YANUAR RISCA PRATIWI AND MUHAMMAD AINUL FIKRI, 2023. Klasifikasi Penyakit Jantung Menggunakan Metode Synthetic Minority Over-Sampling Technique Dan Random Forest Classifier. *Indonesian Journal of Computer Science*, 12(5), pp.2995–3011. <https://doi.org/10.33022/ijcs.v12i5.3413>.
- ANDIE AND HASANUDDIN, 2023. Klasifikasi Tingkat Kompetensi Mahasiswa UNISKA Menggunakan Kombinasi Algoritma K-Nearest Neighbors (KNN) Dan Manhattan Distance. *Technologia : Jurnal Ilmiah*, 14(1), pp.74–77.
- ANGGARA, E.D., WIDJAJA, A. AND SUTEJA,

- B.R., 2022. Prediksi Kinerja Pegawai sebagai Rekomendasi Kenaikan Golongan dengan Metode Decision Tree dan Regresi Logistik. *Jurnal Teknik Informatika dan Sistem Informasi*, 8(1), pp.218–234. <https://doi.org/10.28932/jutisi.v8i1.4479>.
- ARIFIN, T. AND SYALWAH, S., 2020. Prediksi Keberhasilan Immunotherapy Pada Penyakit Kutil Dengan Menggunakan Algoritma Naïve Bayes. *Jurnal Responsif: Riset Sains dan Informatika*, 2(1), pp.38–43. <https://doi.org/10.51977/jti.v2i1.177>.
- DAQIQIL ID, I., 2021. *Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Python*. Riau. <https://doi.org/10.5281/zenodo.5113507>.
- DASTJERDY, B., SAEIDI, A. AND HEIDARZADEH, S., 2023. Review of Applicable Outlier Detection Methods to Treat Geomechanical Data. *Geotechnics*, 3(2), pp.375–396. <https://doi.org/10.3390/geotechnics3020022>.
- GUNAWAN, B., PRATIWI, H.S. AND PRATAMA, E.E., 2018. Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naive Bayes. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 4(2), p.113. <https://doi.org/10.26418/jp.v4i2.27526>.
- GUNTARA, R.G., 2023. Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), pp.55–60.
- HALIM, S.F.N. AND AZMI, U., 2023. Analisis Perbandingan Klasifikasi dan Penerapan Teknik SMOTE Dalam Imbalanced Data Pada Credit Card Default. *Jurnal Sains dan Seni ITS*, 12(2). <https://doi.org/10.12962/j23373520.v12i2.111833>.
- HANCOCK, J.T. AND KHOSHGOFTAAR, T.M., 2020. Survey on categorical data for neural networks. *Journal of Big Data*, [online] 7(1). <https://doi.org/10.1186/s40537-020-00305-w>.
- HERDIAN, C., KAMILA, A. AND AGUNG MUSA BUDIDARMA, I.G., 2024. Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi. *Technologia: Jurnal Ilmiah*, 15(1), p.93. <https://doi.org/10.31602/tji.v15i1.13457>.
- IRYANI, L., 2023. Penerapan Machine Learning Dalam Klasifikasi Kinerja Pegawai Pt X. *Jurnal Informanika*, 09(01), pp.1–6.
- KARO, I.M.K., AMALIA, S.N. AND SEPTIANA, D., 2022. Wildfires Classification Using Feature Selection with K-NN, Naïve Bayes, and ID3 Algorithms. *Information and Communication Technology (SEICT)*, 3(1), pp.15–24.
- KUMALASARI, J.T. AND MERDEKAWATI, A., 2023. Analisis Sentimen Terhadap Program Kampus Merdeka Pada Twitter Menggunakan Metode Naïve Bayes, Union dan Synthetic Minority Over Sampling Technique (SMOTE). *SATIN - Sains dan Teknologi Informasi*, 9(1), pp.01–12. <https://doi.org/10.33372/stn.v9i1.894>.
- KURNIADI, D., NURAENI, F. AND FIRMANSYAH, M., 2022. Klasifikasi Masyarakat Penerima Bantuan Langsung Tunai Dana Desa Menggunakan Naïve Bayes Dan SMOTE. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, x(36), pp.1–11. <https://doi.org/10.25126/jtiik.2023106453>.
- LAGA, S.A., 2023. Perbandingan Metode K-NN dan SVM Berdasarkan Kinerja Pegawai. *Jurnal Sistem Komputer dan Informatika (JSON)*, 4, pp.420–425. <https://doi.org/10.30865/json.v4i3.5816>.
- MOBIUS, 2020. *HR Analytics: Employee Promotion Data*. Kaggle.
- MUTMAINAH, S., 2021. Penanganan Imbalance Data Pada Klasifikasi Kemungkinan Penyakit Stroke. *SNATi*, 1, pp.10–16.
- NIKMATUN, I.A. AND WASPADA, I., 2019. Implementasi Data Mining untuk Klasifikasi Masa Studi Mahasiswa Menggunakan Algoritma K-Nearest Neighbor. *Jurnal SIMETRIS*, 10(2), pp.421–432.
- NUGRAHA, N.P., AZIM, R., DAFFA, S.Z. AND NINGAYU, P.S., 2023. Perbandingan Akurasi Metode Naïve Bayes dan Metode KNN untuk Memprediksi Gagal Ginjal Kronis. *Jurnal Rekayasa Elektro Sriwijaya*, 5(1), pp.1–10. <https://doi.org/10.36706/jres.v5i1.63>.
- NURHOPIAH, A. AND MAGNOLIA, C., 2023. Perbandingan Metode Resampling Pada Imbalanced Dataset Untuk Klasifikasi Komentar Program Mbkm. *Jurnal Publikasi Ilmu Komputer dan Multimedia*, 2(1), pp.9–22. <https://doi.org/10.55606/jupikom.v2i1.862>.
- NURSYAHFITRI, R., ROZIKIN, C. AND ADAM, R.I., 2022. Penerapan Metode SMOTE dalam Klasifikasi Daerah Rawan Banjir di Karawang Menggunakan Algoritma Naive Bayes. *Jurnal Sistem dan Teknologi Informasi (JustIN)*, 10(4), p.339. <https://doi.org/10.26418/justin.v10i4.46935>.
- PERMANA, I. AND SALISAH, F.N.S., 2022. Pengaruh Normalisasi Data Terhadap Performa Hasil Klasifikasi Algoritma Backpropagation. *Indonesian Journal of Informatic Research and Software Engineering (IJIRSE)*, 2(1), pp.67–72. <https://doi.org/10.57152/ijirse.v2i1.311>.
- PRAYOGA, S.A., NAWANGSIH, I. AND WIYATNO, T.N., 2019. Implementasi Metode Naïve Bayes Classifier Untuk Identifikasi Jenis Jamur. *Pelita Teknologi: Jurnal Ilmiah Informatika, Arsitektur dan Lingkungan*, 14(2), pp.134–144.
- RASHED-AL-MAHFUZ, M., HAQUE, A., AZAD, A., ALYAMI, S.A., QUINN, J.M.W. AND

- MONI, M.A., 2021. Clinically Applicable Machine Learning Approaches to Identify Attributes of Chronic Kidney Disease (CKD) for Use in Low-Cost Diagnostic Screening. *IEEE Journal of Translational Engineering in Health and Medicine*, 9(December 2020), pp.1–11. <https://doi.org/10.1109/JTEHM.2021.3073629>.
- REGINA, S., SUTINAH, E. AND AGUSTINA, N., 2021. Clustering Kualitas Kinerja Karyawan Pada Perusahaan Bahan Kimia Menggunakan Algoritma K-Means. *Jurnal Media Informatika Budidarma*, 5(2), p.573. <https://doi.org/10.30865/mib.v5i2.2909>.
- SIHAHAN, M., 2021. An Analysis of Contract Employee Performance Assessment Using Machine Learning. *JITE (Journal of Informatics and Telecommunication Engineering)*, 5(1).
- SIHOMBING, P.R., SURYADININGRAT, SUNARJO, D.A. AND YUDA, Y.P.A.C., 2023. Identifikasi Data Outlier (Pencilan) dan Kenormalan Data Pada Data Univariat serta Alternatif Penyelesaiannya. *Jurnal Ekonomi Dan Statistik Indonesia*, 2(3), pp.307–316. <https://doi.org/10.11594/jesi.02.03.07>.
- SIRINGORINGO, R., 2018. Klasifikasi Data Tidak Seimbang Menggunakan Algoritma Smote Dan K-Nearest Neighbor. *Journal Information System Development (ISD)*, 3(1), pp.44–49.
- SOTARJUA, L.M. AND SANTOSO, D.B., 2022. Perbandingan Algoritma Knn, Decision Tree, Dan Random Forest Pada Data Imbalanced Class Untuk Klasifikasi Promosi Karyawan. *Jurnal INSTEK (Informatika Sains dan Teknologi)*, 7(2), pp.192–200. <https://doi.org/10.24252/instek.v7i2.31385>.
- SUMANTRI, G., NOVIANTO, M.D. AND PRIHASTUTI, P.P., 2023. Implementasi Fuzzy C-Means dalam Pengelompokan Provinsi di Indonesia untuk Pemerataan Kualitas Pendidikan. *Prosiding Seminar Pendidikan Matematika dan Matematika*, 8(2721). <https://doi.org/10.21831/pspmm.v8i2.310>.
- SUSANA, H., SUARNA, N., FATHURROHMAN AND KASLANI, 2022. Penerapan Model Klasifikasi Metode Naive Bayes Terhadap Penggunaan Akses Internet. *Jurnal Riset Sistem Informasi dan Teknologi Informasi (JURSISTEKNI)*, 4(1), pp.1–8. <https://doi.org/10.52005/jursistekni.v4i1.96>.
- VIRANTIKA, E., KUSNAWI, K. AND IPMAWATI, J., 2022. Evaluasi Hasil Pengujian Tingkat Clusterisasi Penerapan Metode K-Means Dalam Menentukan Tingkat Penyebaran Covid-19 di Indonesia. *Jurnal Media Informatika Budidarma*, 6(3), p.1657. <https://doi.org/10.30865/mib.v6i3.4325>.
- WARING, J., LINDVALL, C. AND UMETON, R., 2020. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine*, 104(January), p.101822. <https://doi.org/10.1016/j.artmed.2020.101822>.
- WIRAYASA, I.K.A. AND SANTOSO, H., 2022. Analisis Employee Satisfaction Menggunakan Teknik Clustering Dan Classification Machine Learning. *Progresif: Jurnal Ilmiah Komputer*, 18(1), p.1. <https://doi.org/10.35889/progresif.v18i1.766>.
- YULIAN PAMUJI, F., AHMAD ROFIQUL MUSLIKH, RIZZA MUHAMMAD ARIEF AND DELVIANA MUTI, 2024. Komparasi Metode Mean dan KNN Imputation dalam Mengatasi Missing Value pada Dataset Kecil. *Jurnal Informatika Polinema*, 10(2), pp.257–264. <https://doi.org/10.33795/jip.v10i2.5031>.

Halaman ini sengaja dikosongkan.