

## Komparasi Algoritma Machine Learning dalam Memprediksi Penyakit Gagal Ginjal

\* Wahyu Ardiantito S

Universitas Negeri Medan

Rizki Agung Ramadhan

Universitas Negeri Medan

Richard Steven Immanuel S

Universitas Negeri Medan

Alamat: Kampus: Jl. William Iskandar Ps. V, Kenangan Baru, Kec. Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara 20221

\*Korespondensi penulis: [ardiantitowahyu@gmail.com](mailto:ardiantitowahyu@gmail.com)

**Abstract.** *Chronic Kidney Disease (CKD) is a serious health problem, with significant impact on patients' quality of life and healthcare costs. In an effort to improve early diagnosis, a comparison was made between several Machine Learning algorithms used for analysis of patient clinical data. This clinical data contains the medical history or health records of patients. The Machine Learning algorithms used in this study include K Nearest Neighbor (KNN), Support Vector Machine (SVM), and Logistic Regression. By searching for the best algorithm through the calculation of Accuracy, Precision, and Recall with comparison when using SMOTE (Synthetic Minority Oversampling) to balancing the class attribute.*

**Keywords:** PGGK, K Nearest Neighbor, Support Vector Machine, SMOTE

**Abstrak.** Penyakit Gagal Ginjal Kronis (PGGK) adalah sebuah penyakit dengan tingkat keseriusan yang tinggi dan mempunyai dampak yang signifikan terhadap kualitas hidup pasien beserta biaya pengobatannya. Dalam upaya untuk meningkatkan diagnosis dini dilakukan komparasi antara beberapa algoritma Machine Learning yang digunakan untuk analisis data klinis pasien. Data klinis ini merupakan data yang memuat riwayat ataupun catatan kesehatan pasien. Algoritma Machine Learning pada penelitian ini menggunakan K Nearest Neighbor (KNN), Support Vector Machine (SVM), dan Logistic Regression. Dengan pencarian algoritma terbaik melalui perhitungan Akurasi, Presisi, dan Recall beserta perbandingannya apabila dilakukan penyetaraan kelas dengan teknik SMOTE (Teknik Oversampling Minoritas)

**Kata kunci:** PGGK, K Nearest Neighbor, Support Vector Machine, SMOTE

### LATAR BELAKANG

Gagal Ginjal Kronis (GGK) merupakan masalah pada fungsi ginjal yang bersifat semakin memburuk dan tidak dapat disembuhkan, Penyakit ini memiliki dampak di mana tubuh kehilangan kemampuan untuk mempertahankan metabolisme dan keseimbangan cairan serta elektrolit. Hal ini menyebabkan peningkatan kadar ureum dalam tubuh. Penyebab GGK antara lain kondisi metabolic seperti diabetes melitus, infeksi seperti pielonefritis, obstruksi saluran kemih, gangguan imunologis, hipertensi, kerusakan tubulus ginjal akibat zat nefrotoksin, dan kelainan bawaan ginjal yang menurunkan laju filtrasi glomerulus (GFR). Penderita GGK memiliki kondisi yang persisten, tidak dapat disembuhkan, dan memerlukan pengobatan jangka panjang berupa transplantasi ginjal, dialisis peritoneal, hemodialisis, serta perawatan non-bedah secara rutin. (Narsa, A.C, *et al*, 2022).

Machine learning (ML) adalah bagian dari kecerdasan buatan (AI) yang memungkinkan komputer melakukan tugas tertentu tanpa instruksi eksplisit. Ketika digunakan dalam pemodelan prediktif, algoritma ML dapat dilatih untuk menangkap pola-pola yang mendasari data sampel dan membuat prediksi tentang data baru berdasarkan informasi yang diperoleh. Dengan adanya model prediksi yang akurat, dokter dan tenaga medis dapat mengidentifikasi pasien yang berisiko tinggi mengalami gagal ginjal dan mengambil tindakan pencegahan atau pengobatan yang tepat. Selain itu, penelitian lebih lanjut dalam bidang ini dapat membantu dalam pemahaman lebih lanjut tentang faktor-faktor yang berkontribusi terhadap penyakit ini dan pengembangan strategi penanganan yang lebih efektif.

Beberapa penelitian terdahulu juga pernah melakukan perbandingan algoritma machine learning dalam memprediksi penyakit. Penelitian mengenai komparasi machine learning terhadap penyakit jantung menggunakan SVM, ANN, dan Logistic Regression menghasilkan model logistic regression sebagai model dengan akurasi tertinggi yaitu 80% (F. Handayani, 2021). Penelitian lain melakukan komparasi model SVM dan XGBOOST untuk kanker payudara memperoleh akurasi sebesar 97,58% untuk SVM dan XGBOOST sebesar 96,26%. (M. Ravly, *et al*, 2022)

Dengan menggabungkan tiga algoritma ini, penelitian ini akan menganalisis kinerja masing-masing dalam memprediksi penyakit gagal ginjal. Hasil dari komparasi ini diharapkan memberikan wawasan tentang algoritma yang paling akurat dan efektif dalam melakukan prediksi penyakit tersebut berdasarkan dataset yang diambil.

## **KAJIAN TEORITIS**

### **1. Klasifikasi**

Klasifikasi berasal dari kata Latin "classis," yang merujuk pada proses pengelompokan. Dalam bahasa Inggris, istilah "classification" mengindikasikan penggolongan atau pembagian. Di dalam konteks bahasa Indonesia, klasifikasi mengacu pada tindakan mengelompokkan sesuatu berdasarkan jenisnya. Proses klasifikasi melibatkan penilaian terhadap objek data untuk menempatkannya dalam kategori tertentu dari sejumlah kategori yang ada. Dalam melaksanakan klasifikasi, suatu model dibangun berdasarkan data latih yang telah ada, dan model tersebut kemudian digunakan untuk mengklasifikasikan data baru. (Utomo, D.P., *et al*, 2020)

Definisi lebih lanjut mengenai klasifikasi adalah kegiatan yang melibatkan pelatihan atau pembelajaran terhadap atribut target yang memetakan setiap atribut fitur ke satu label kelas

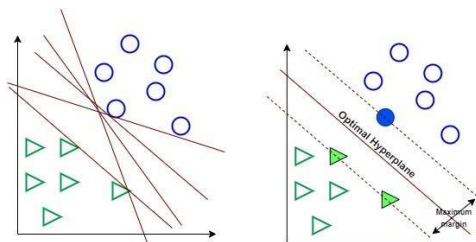
yang tersedia. Meskipun diharapkan bahwa sistem klasifikasi dapat mengklasifikasikan semua data set dengan akurat, realitasnya adalah kinerja sistem tidak dapat mencapai tingkat keakuratan 100%. Oleh karena itu, evaluasi kinerja sistem klasifikasi perlu dilakukan. Tujuan klasifikasi melibatkan upaya untuk menemukan kembali bahan pustaka dalam perpustakaan, tanpa memandang seberapa besar atau kecil koleksi perpustakaan tersebut. Dalam konteks yang lebih spesifik, tujuan klasifikasi perpustakaan mencakup (Handayanto, A, *et al*, 2019):

- a. Menghasilkan urutan yang bermanfaat
- b. Penempatan koleksi yang tepat
- c. Mempermudah mekanisme penyusunan bahan Pustaka
- d. Mempermudah penempatan koleksi baru (tambahan koleksi)
- e. mempermudah penemuan bahan Pustaka

## 2. SVM

SVM merupakan sebuah bagian dari pembelajaran mesin yang bekerja dengan cara melakukan pengamatan data dan mengidentifikasi pola-pola yang telah ada. Metode ini umumnya digunakan pada penyelesaian kasus klasifikasi. Namun, kemampuannya tidak terbatas pada itu saja; SVM juga dapat melakukan estimasi dan evaluasi terhadap suatu sistem. Tujuan utama dari SVM yaitu mendapatkan nilai berdasarkan frekuensi munculnya suatu variabel dan kemudian mengkategorikannya sebagai kelas positif dan negatif. SVM dikembangkan oleh Vapnik, Guyon, dan Boser, dan pertama kali diperkenalkan pada sekitar tahun 1992 dalam Annual Workshop on Computational Learning Theory. Metode ini merupakan mesin pembelajaran dengan tujuan mencari hyperplane optimal yang dapat memisahkan dua kelas dalam input space.

Beberapa bidang penelitian terkait dengan algoritma SVM mencakup implementasi data mining dengan model SVM untuk memprediksi kepuasan pengunjung taman tabebuya. Hasil penelitian ini menghasilkan validasi dan evaluasi, menyimpulkan bahwa algoritma SVM memiliki kinerja dan akurasi rata-rata sebesar 86%, dengan nilai AUC mencapai 0.947.



**Gambar 1. SVM**

Beberapa pola data terkelompok dalam 2 kelas yaitu kelas 1 (warna merah) dan kelas +1 (warna kuning). Terdapat 2 kasus pemisahan kelas oleh Hyperplane. Kelas yang dapat

dipisahkan dengan sempurna oleh Hyperplane disebut sebagai Support Vector Machine (SVM) linier. Sedangkan kelas yang tidak dapat dipisahkan dengan sempurna oleh Hyperplane disebut sebagai SVM non-linier. Pada dasarnya, SVM non-linier merupakan solusi dari permasalahan SVM linier dengan menerapkan fungsi kernel pada ruang fitur berdimensi tinggi. Persamaan matematis SVM linier dan non-linier dapat dilihat pada Tabel 1. (Handayani, F., *et al* 2021 ).

JENIS SVM	Tipe Kernel	Formula
SVM Linear	Linier	$K(x,y) = x,y$
SVM Nonlinear	Polynomial	$K(x,y) = (x,y + 1)^p$
	Gaussian Radial Basis Function	$K(x,y) = e^{-\frac{ x-y ^2}{2\sigma^2}}$
	Sigmoid	$(x,y) = \tanh(Kx.y - \delta)$

**Tabel 1. Jenis Kernel SVM**

Keterangan :

P: pangkat (degree of)

$\sigma$ : nilai sigma

$\delta$ : nilai delta

### 3. KNN

K-Nearest Neighbor (KNN) menonjol sebagai salah satu algoritma sederhana yang digunakan untuk mengatasi tantangan klasifikasi, yang sering diterapkan dalam kategorisasi teks dan data. Pendekatan ini memerlukan pengklasifikasian objek berdasarkan kedekatan data dengan objek masing-masing. Diklasifikasikan sebagai bentuk pembelajaran terawasi, algoritma KNN menetapkan objek data baru ke dalam kategori berdasarkan kedekatan terdekat dalam hal jarak. Algoritma K-Nearest Neighbor (K-NN) adalah metode untuk mengklasifikasikan objek berdasarkan data pembelajaran/latih yang paling dekat jaraknya dengan objek tersebut. K-NN mencari kasus yang paling mirip dengan kasus baru yang akan diklasifikasikan. Kedekatan antar kasus diukur berdasarkan kecocokan nilai beberapa fitur yang ada pada masing-masing kasus. Untuk menghitung jarak antara dua titik, yaitu titik pada data latih dan titik pada data uji, umumnya menggunakan rumus Euclidean seperti yang ditunjukkan pada persamaan (1).

$$(x,y) = \sqrt{\sum(x_k - y_k)^2} \dots \dots \dots (1)$$

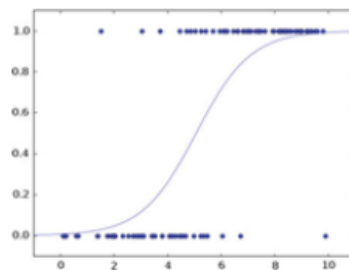
(x,y) merupakan jarak antara titik pada data variable x dengan data pada variable y yang akan diklasifikasi. x dan y masing-masing memiliki komponen x1, x2, ..., dst dan y1, y2, ..., dst. Dimana i mewakili nilai dari data pada atribut dan n merupakan jumlah total atribut. Pada proses pelatihan, algoritma K-NN akan menyimpan vektor-vektor atribut dan label kelas dari data training untuk dipelajari. Pada fase klasifikasi, atribut yang sama dihitung untuk data uji yang kelasnya tidak diketahui. Kemudian dihitung jarak sebuah vektor pada data uji dengan

keseluruhan vektor yang telah disimpan pada data latih lalu digunakan sebanyak k buah tetangga terdekat.”( Harun, R., *et al* 2020).

#### 4. Logistic Regression

Regresi Logistik memerlukan pembelajaran yang diawasi untuk tantangan regresi dan klasifikasi. Ini memanfaatkan probabilitas untuk mengantisipasi kategorisasi data dengan atribut kategorikal. Penggabungan nilai masukan dilakukan secara linier dengan memanfaatkan fungsi sigmoid beserta nilai koefisien untuk meramalkan hasilnya. Estimasi kemungkinan maksimum digunakan melalui fungsi sigmoid untuk mengasumsikan data yang paling mungkin, dengan menetapkan probabilitas mulai dari 0 hingga 1 untuk menunjukkan kemungkinan terjadinya suatu peristiwa. Memperkenalkan ambang batas keputusan mengubahnya menjadi masalah klasifikasi. Klasifikasi tersebut dapat mencakup Biner, Multinomial, atau Ordinal. Hal ini menyajikan model implementasi yang mudah dan berpotensi menghasilkan prediksi yang akurat.

$$p = \frac{1}{1 + e^{-(a+bX)}} \quad (2)$$



**Gambar 2. Logistik regresi**

dimana dalam Persamaan (2) p adalah probabilitas dan a, b akan menjadi parameter model, X adalah faktor.

Algoritma Regresi Logistik (Logistic Regression) adalah metode regresi dan klasifikasi untuk mengevaluasi dataset yang berisi satu atau lebih variabel bebas (independen) untuk menarik suatu kesimpulan/hasil. Hasil tersebut diukur dengan variabel terikat (dependen) yang bersifat dikotomi (hanya ada dua kemungkinan nilai).

#### 5. Matriks Konfusi

Matriks konfusi, biasa disebut sebagai matriks kesalahan, berfungsi sebagai representasi tabel yang digunakan untuk menilai efektivitas model klasifikasi ketika diterapkan pada kumpulan data pengujian yang nilai sebenarnya sudah teridentifikasi. Konsep mendasar di balik matriks kebingungan adalah membandingkan hasil yang diharapkan dengan data asli, yang sering disebut sebagai kebenaran dasar.

Matriks ini mencakup empat kombinasi berbeda antara hasil yang diproyeksikan dan nilai faktual, khususnya True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN).

Setelah mendapatkan nilai TP, TN, FP, dan FN, langkah berikutnya adalah menghitung akurasi, sensitivitas, dan spesifisitas untuk mengukur kinerja dari sistem klasifikasi yang telah dibuat. Berikut adalah rumus-rumus untuk menghitung nilai akurasi, sensitivitas, dan spesifisitas.

a. Akurasi

Akurasi adalah metrik yang digunakan untuk mengukur sejauh mana sistem klasifikasi yang telah dibangun mampu melakukan klasifikasi secara akurat, baik untuk data yang memiliki label positif maupun negatif. Semakin tinggi nilai akurasi yang diperoleh, semakin baik sistem klasifikasi tersebut dalam melakukan prediksi (Miladitiya, 2018). Nilai maksimal akurasi adalah 100%. Formula untuk menghitung akurasi dalam kasus klasifikasi dengan beberapa kelas (multi-class) dapat ditemukan dalam Persamaan (3).

$$ACC = \frac{TP_{all}}{N_{all}} \times 100\% \quad (3)$$

di mana  $TP_{all}$  yaitu total nilai True Positive pada seluruh kelas dan  $N_{all}$  = total seluruh data.

b. Sensitivitas

Sensitivitas menggambarkan seberapa baik sistem klasifikasi mampu mengidentifikasi data yang sebenarnya bernilai positif dan mengklasifikasikannya ke dalam kelas positif. Semakin tinggi nilai sensitivitas yang diperoleh, semakin baik sistem klasifikasi dalam mengenali objek yang memiliki kondisi positif atau penyakit (Miladitiya, 2018). Batas maksimal sensitivitas adalah 100%. Rumus untuk menghitung sensitivitas dalam konteks klasifikasi multi-kelas dapat ditemukan dalam Persamaan (4).

$$SEN = \frac{\left(\frac{TP}{TP+FN}\right)}{N_{all}} \times 100\% \quad (4)$$

Dimana n merupakan seluruh data.

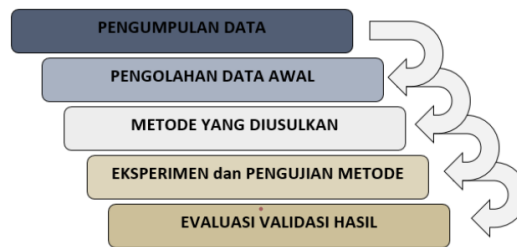
## METODE PENELITIAN

### 1. Tahapan Penelitian

Penelitian ini dimulai dengan mengambil data dari situs UCI Machine Learning Repository. Data tersebut kemudian diolah menjadi data awal yang akan diproses

menggunakan Jupyter Notebook. Selanjutnya, data diuji dengan berbagai algoritma menggunakan dataset yang sama atau objek yang sama.

Tiga algoritma yang akan dibandingkan dalam penelitian ini adalah KNN, Support Vector Machine, dan Logistic Regression. Setelah data diproses dan diuji, langkah selanjutnya adalah melakukan analisis terhadap data yang dihasilkan oleh ketiga algoritma machine learning di Jupyter Notebook. Tujuannya adalah untuk menentukan algoritma yang paling optimal untuk diterapkan pada data yang telah diuji. Gambar dibawah ini merupakan tahapan penelitian yang akan dilakukan



**Gambar 3. Tahapan Penelitian**

## 2. Data yang digunakan

Dataset yang digunakan dalam penelitian komparasi algoritma machine learning untuk memprediksi penyakit gagal ginjal kronis diambil dari situs web UCI Machine Learning Repository. Situs ini merupakan platform yang menyediakan akses ke berbagai dataset yang dapat digunakan untuk keperluan penelitian. Dataset yang digunakan dalam penelitian ini diperoleh dari berbagai rumah sakit Karaikudi, Tamilnadu, India dan memiliki 400 data. Terdiri atas 25 atribut dimana 11 atribut bertipe data numerik dan 14 atribut bertipe nominal. Gambar 2 menunjukkan atribut atribut dari dataset yang digunakan.

S. No.	Attribute	Description
1	Age (numerical)	Age in years
2	Blood Pressure (numerical)	bp in mm/Hg
3	Specific Gravity (nominal)	Sg-(1.005,1.010,1.015,1.020,1.025)
4	Albumin (nominal)	al - (0,1,2,3,4,5)
5	Sugar (nominal)	su - (0,1,2,3,4,5)
6	Red Blood Cells (nominal)	rbc - (normal, abnormal)
7	Pus Cell (nominal)	pc - (normal, abnormal)
8	Pus Cell clumps (nominal)	pcc - (present, notpresent)
9	Bacteria (nominal)	ba - (present, notpresent)
10	Blood Glucose Random (numerical)	bgr in mgs/dl
11	Blood Urea (numerical)	bu in mgs/dl
12	Serum Creatinine (numerical)	sc in mgs/dl
13	Sodium (numerical)	sod in mEq/L
14	Potassium (numerical)	pot in mEq/L
15	Haemoglobin (numerical)	hemo in gms
16	Packed Cell Volume (numerical)	pcv
17	White Blood Cell Count (numerical)	wc in cells/cumm
18	Red Blood Cell Count (numerical)	rc in millions/cmm
19	Hypertension (nominal)	htn - (yes, no)
20	Diabetes Mellitus (nominal)	dm - (yes, no)
21	Coronary Artery Disease (nominal)	cad - (yes, no)
22	Appetite (nominal)	appet - (good, poor)
23	Pedal Edema (nominal)	pe - (yes, no)
24	Anemia (nominal)	ane - (yes, no)
25	Class (nominal)	class - (ckd, notckd)

**Gambar 4. Atribut Dataset**

### 3. Tahap preprocessing

Pada tahap preprocessing dilakukan beberapa hal yaitu pengisian nilai missing value, melakukan transformasi data, dan melakukan normalisasi dengan z-score. Setelah dilakukan tahap preprocessing kemudian data dibagi 2 yaitu variable bebas dan variable terikat. Variabel bebas adalah semua variable pada data tanpa atribut "class", dan variable terikat adalah data "class tersebut".

### 4. Evaluasi Model

Tahapan berikutnya adalah melakukan evaluasi model dengan beberapa metode yang digunakan untuk melihat algoritma apa yang bekerja dengan baik pada dataset penelitian ini. Berikutnya untuk menambah perbandingan juga dilakukan teknik SMOTE untuk menyeimbangkan jumlah kelas.

## HASIL DAN PEMBAHASAN

Pemodelan menggunakan algoritma KNN, SVM, dan Logistic Regression juga dilakukan perhitungan nilai akurasi presisi, f-1, dan recall beserta matriks konfusinya untuk melihat seberapa baik algoritma ini digunakan. Kemudian menentukan algoritma mana yang memiliki performa paling baik pada data yang digunakan

### 1. Model menggunakan KNN

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

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

knn_acc = accuracy_score(y_test, knn.predict(X_test))

print(f"Training Accuracy of KNN is {accuracy_score(y_train, knn.predict(X_train))}")
print(f"Test Accuracy of KNN is {knn_acc} \n")

print(f"Confusion Matrix :- \n{confusion_matrix(y_test, knn.predict(X_test))}\n")
print(f"Classification Report :- \n {classification_report(y_test, knn.predict(X_test))}")
```

**Gambar 5.** Model KNN Menggunakan Python

Gambar diatas menunjukkan proses pemodelan menggunakan algoritma knn menggunakan bahasa pemrograman python. Proses yang dilakukan pada script diatas adalah memanggil library, mendefinisikan variable yang memuat fungsi knn, memasukkan data train ke model knn dan melakukan perhitungan akurasi beserta matriks konfusinya.

```

Training Accuracy of KNN is 0.7928571428571428
Test Accuracy of KNN is 0.725

Confusion Matrix :-
[[46 26]
 [ 7 41]]

Classification Report :-
              precision    recall  f1-score   support

     0       0.87       0.64       0.74       72
     1       0.61       0.85       0.71       48

 accuracy          0.74          0.75          0.73       120
 macro avg         0.74          0.75          0.72       120
 weighted avg     0.77          0.72          0.73       120

```

**Gambar 6.** Evaluasi Model KNN

Gambar 6 menunjukkan akurasi yang dihasilkan menggunakan algoritma KNN yaitu 79% akurasi untuk data training dan 72% akurasi untuk data test. Matriks konfusi menggambarkan bahwa sebesar 46 data berhasil diprediksi dengan benar untuk class “0(Gagal Ginjal)”, 26 data class “0(Gagal Ginjal)” diprediksi sebagai class “1(Tidak Gagal Ginjal)”, 41 data berhasil diprediksi dengan benar untuk class “1( Tidak Gagal Ginjal)”, dan 7 data class “1( Tidak Gagal Ginjal)” diprediksi sebagai class “0(Gagal Ginjal)”.

## 2. Model Menggunakan SVM

```

from sklearn.svm import SVC
svm = SVC()
svm.fit(X_train, y_train)

# accuracy score, confusion matrix and classification report of svm

svm_acc = accuracy_score(y_test, svm.predict(X_test))

print(f"Training Accuracy of SVM is {accuracy_score(y_train, svm.predict(X_train))}")
print(f"Test Accuracy of SVM is {svm_acc} \n")

print(f"Confusion Matrix :- \n{confusion_matrix(y_test, svm.predict(X_test))}\n")
print(f"Classification Report :- \n {classification_report(y_test, svm.predict(X_test))}")

```

**Gambar 7.** Model SVM Menggunakan Python

Gambar 7 menunjukkan proses pemodelan menggunakan algoritma svm menggunakan bahasa pemrograman python. Proses yang dilakukan pada script diatas adalah memanggil library, mendefinisikan variable yang memuat fungsi svm, memasukkan data train ke model knn dan melakukan perhitungan akurasi beserta matriks konfusinya.

```

Training Accuracy of SVM is 0.6428571428571429
Test Accuracy of SVM is 0.6

Confusion Matrix :-
[[62 10]
 [38 10]]

Classification Report :-
              precision    recall  f1-score   support

     0       0.62       0.86       0.72       72
     1       0.50       0.21       0.29       48

 accuracy          0.56          0.53          0.60       120
 macro avg         0.56          0.53          0.51       120
 weighted avg     0.57          0.60          0.55       120

```

**Gambar 8.** Evaluasi Model SVM

Gambar 8 menunjukkan akurasi yang dihasilkan menggunakan algoritma KNN yaitu 64% akurasi untuk data training dan 60% akurasi untuk data test. Matriks konfusi menggambarkan bahwa sebesar 62 data berhasil diprediksi dengan benar untuk class “0(Gagal Ginjal)”, 10 data class “0(Gagal Ginjal)” diprediksi sebagai class “1(Tidak Gagal Ginjal)”, 10 data berhasil diprediksi dengan benar untuk class “1( Tidak Gagal Ginjal)”, dan 38 data class “1( Tidak Gagal Ginjal)” diprediksi sebagai class “0(Gagal Ginjal)”.

### 3. Model Menggunakan Logistik Regression

```

from sklearn.linear_model import LogisticRegression
logres = LogisticRegression()
logres.fit(X_train, y_train)

# accuracy score, confusion matrix and classification report of logistic regression

logres_acc = accuracy_score(y_test, logres.predict(X_test))

print(f"Training Accuracy of Logistic Regression is {accuracy_score(y_train, logres.predict(X_train))}")
print(f"Test Accuracy of Logistic Regression is {logres_acc} \n")

print(f"Confusion Matrix :- \n{confusion_matrix(y_test, logres.predict(X_test))}\n")
print(f"Classification Report :- \n{classification_report(y_test, logres.predict(X_test))}")
    
```

**Gambar 9.** Model Logistic Regression Menggunakan Python

Gambar 9 menunjukkan proses pemodelan menggunakan algoritma logistic regression menggunakan bahasa pemrograman python. Proses yang dilakukan pada script diatas adalah memanggil library, mendefinisikan variable yang memuat fungsi logistic regression, memasukkan data train ke model knn dan melakukan perhitungan akurasi beserta matriks konfusinya.

```

Training Accuracy of Logistic Regression is 0.8964285714285715
Test Accuracy of Logistic Regression is 0.9

Confusion Matrix :-
[[68  4]
 [ 8 40]]

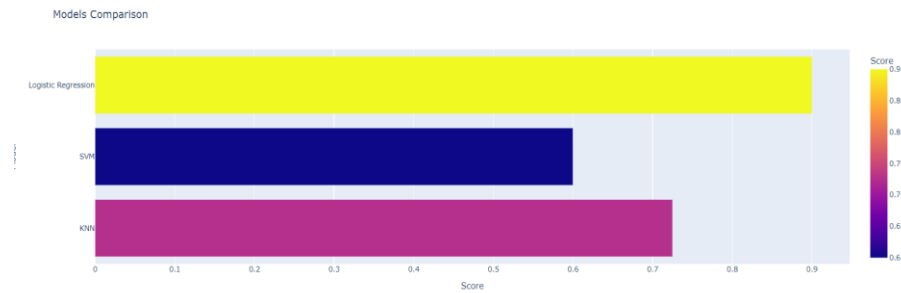
Classification Report :-
              precision    recall  f1-score   support

     0       0.89        0.94        0.92         72
     1       0.91        0.83        0.87         48

 accuracy          0.90
 macro avg          0.90
 weighted avg       0.90
    
```

**Gambar 10.** Evaluasi Model Logistic Regression

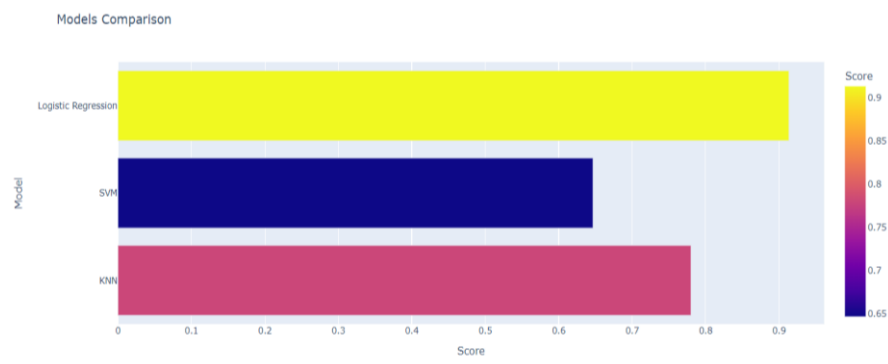
Gambar 10 menunjukkan akurasi yang dihasilkan menggunakan algoritma KNN yaitu 89% akurasi untuk data training dan 90% akurasi untuk data test. Matriks konfusi menggambarkan bahwa sebesar 68 data berhasil diprediksi dengan benar untuk class “0(Gagal Ginjal)”, 4 data class “0(Gagal Ginjal)” diprediksi sebagai class “1(Tidak Gagal Ginjal)”, 40 data berhasil diprediksi dengan benar untuk class “1( Tidak Gagal Ginjal)”, dan 8 data class “1( Tidak Gagal Ginjal)” diprediksi sebagai class “0(Gagal Ginjal)”.



**Gambar 11.** Komparasi Model

Dari ketiga algoritma yang digunakan (knn, svm, logistic regression) didapatkan hasil bahwa algoritma logistic regression memiliki performa dan akurasi yang paling baik dalam mengklasifikasi dan memprediksi penyakit gagal ginjal kronis dengan akurasi sebesar 90%, kemudian diikuti algoritma knn dengan akurasi sebesar 73%, dan algoritma svm dengan akurasi terkecil yaitu sebesar 60%.

Berikutnya menggunakan teknik SMOTE untuk menyeimbangkan jumlah kelas didapatkan akurasi setiap model didapatkan hasil yaitu knn sebesar 78%, svm sebesar 63%, dan regresi logistic sebesar 92%. Hasil ini menggambarkan bahwa penyeimbangan kelas dapat mempengaruhi akurasi model untuk meningkatkan akurasi.



**Gambar 12.** Komparasi Model Setelah SMOTE

## KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian yang dilakukan dengan algoritma K-Nearest Neighbors (KNN), Support Vector Machine (SVM), dan Logistic Regression, dapat disimpulkan bahwa Logistic Regression memberikan hasil terbaik dalam hal akurasi dan kinerja untuk memprediksi dataset penyakit gagal ginjal kronis pada penelitian ini, baik menggunakan ataupun tanpa SMOTE. Algoritma ini mencapai akurasi sebesar 92%, sedangkan KNN dan SVM hanya mencapai akurasi 78% dan 63% masing-masing dengan menggunakan penyeimbangan jumlah kelas.

## DAFTAR REFERENSI

- A. C. Narsa, V. Maulidya, D. Reggina, W. Andriani, and H. R. Rijai, “Studi Kasus: Pasien Gagal Ginjal Kronis (Stage V) dengan Edema Paru dan Ketidakseimbangan Cairan Elektrolit,” *Jurnal Sains dan Kesehatan*, vol. 4, no. SE-1, pp. 17–22, Dec. 2022, doi: 10.25026/jsk.v4ise-1.1685
- M. Ravly Andryan *et al.*, “KOMPARASI KINERJA ALGORITMA XGBOOST DAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) UNTUK DIAGNOSA PENYAKIT KANKER PAYUDARA,” *Jurnal Informatika dan Komputer*, vol. 6, no. 1, pp. 1–5, 2022.
- Utomo, D.P., Mesran., “Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut pada Data Set Penyakit Jantung,” *Jurnal Media Informatika Budidarma*, vol. 4, no 2, pp 437-444, 2020.
- Handayanto, A., Latifa, K., Saputro, N., D., Waliyansyah, R., R., “Analisis dan Penerapan Algoritma Support Vector Machine (SVM) dalam Data Mining untuk Menunjang Strategi Promosi.” *JUITA: Jurnal Informatika*, vol.7, no.2. 2019
- Amalia, A., Zaidiah, A., Isnainiyah, I., N., “Prediksi Kualitas Udara Menggunakan Algoritma K-Nearest Neighbor”
- Harun, R., Pelangin, Lasena, Y., “Penerapan Data Mining Untuk Menentukan Potensi Hujan Harian Dengan Menggunakan Algoritma K-Nearest Neighbor(KNN).” *MISI (Jurnal Manajemen Informatika & Sistem Informasi)*, vol.3, no.1. 2020
- Handayani, F., Kusuma, K., S., Asbudi, H., L., Purnasiwi, R., G., Kusuma, R., Sunyoto, A., Pradnya, W., M., “Komparasi Support Vector Machine, Logistic Regression dan Artificial Neural Network dalam Prediksi Penyakit Jantung.” *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 7, no. 3. 2021
- Utami T. M., “Klasifikasi Kanker Kulit Pada Citra Dermoscopy Menggunakan Gray Level Co-Occurrence Matrix (GlcM) Dan Kernel Extreme Learning Machine (Kelm)” 2022