

# Perbandingan Akurasi Metode Naïve Bayes dan Metode KNN untuk Memprediksi Gagal Ginjal Kronis

Nandi Prabu Nugraha<sup>1</sup>, Rafian Azim<sup>1</sup>, Syauqi Zalffa Daffa<sup>1</sup>, Putri Salma Ningayu<sup>1</sup>

<sup>1</sup>Teknik Elektro, Fakultas Teknik  
Universitas Sriwijaya, Indonesia

Penulis korespondensi: nandiprabu9207@gmail.com,  
rafianazim6@gmail.com, syauqizalffa14@gmail.com,  
putrisalma2002@gmail.com

**Abstrak**— Saat ini banyak penderita penyakit gagal ginjal kronis bukan hanya dialami oleh orang tua, akan tetapi dialami oleh anak-anak. Tak sedikit yang memakan korban jiwa dalam penyakit ini. Gagal ginjal merupakan kasus dimana ginjal tidak dapat berfungsi dengan baik dan tidak bekerja sebagaimana mestinya. Gejala yang muncul pada penyakit ini muncul secara bertahap, akan tetapi gejala-gejala ini biasanya tidak nampak jelas, yang mengakibatkan penurunan fungsi ginjal seringkali tidak terlalu nampak, sehingga ketika tiba di tahap yang parah, penderita gagal ginjal hanya tinggal menunggu kematian. Langkah pertama dalam pengelolaan penyakit ginjal adalah penetapan diagnosis yang tepat sehingga perlunya perbandingan akurasi antar metode yang digunakan. Banyak metode yang telah digunakan untuk memprediksi penyakit gagal ginjal kronis. Terdapat 2 metode yaitu Naïve Bayes Classifier lebih tepat diterapkan pada data yang besar dan dapat menangani data yang tidak lengkap (missing value) serta kuat terhadap atribut yang tidak relevan dan noise pada data. Sedangkan metode K-Nearest Neighbors yang metodenya mudah untuk dipelajari dan metode yang dipakai sederhana. Dari hasil penelitian Metode K-Nearest Neighbors memiliki akurasi lebih besar dari Naive Bayes Classification sebesar 96%. Sedangkan Naive Bayes Classification memiliki akurasi confusion matrix sebesar 90.00%.

**Kata Kunci** — *penyakit gagal ginjal, naive bayes, KNN*

**Abstract**— Currently, many sufferers of chronic kidney failure are not only experienced by parents, but also experienced by children. Not a few who took lives in this disease. Kidney failure is a case where the kidneys cannot function properly and do not work as they should. Symptoms that appear in this disease appear gradually, but these symptoms are usually not obvious, which results in a decrease in kidney function that is often not very visible, so that when it arrives at a severe stage, patients with kidney failure are just waiting for death. The first step in the management of kidney disease is establishing the right diagnosis, so it is necessary to compare the accuracy between the methods used. Many methods have been used to predict chronic kidney failure. There are 2 methods, namely the Naïve Bayes Classifier which is more appropriate to apply to large data and can handle incomplete data (missing values) and is strong against irrelevant attributes and noise in the data. While the K-Nearest Neighbors method is an easy method to learn and the method used is simple. From the research results, the K-Nearest Neighbors method has greater accuracy than the Naive Bayes Classification of 96%. Meanwhile, Naive Bayes Classification has a confusion matrix accuracy of 90.00%.

**Keywords**— *chronic kidney disease, naive bayes, KNN*

## I. PENDAHULUAN

Penyakit gagal ginjal kronis merupakan permasalahan kesehatan masyarakat dengan populasi yang setiap tahunnya selalu meningkat. Gagal ginjal merupakan kasus dimana ginjal tidak dapat berfungsi dengan baik dan tidak bekerja sebagaimana mestinya. Gejala yang muncul pada penyakit ini muncul secara bertahap, akan tetapi gejala-gejala ini biasanya tidak nampak jelas, yang mengakibatkan penurunan fungsi ginjal seringkali tidak terlalu nampak, sehingga ketika tiba di tahap yang parah, penderita gagal ginjal hanya tinggal menunggu kematian [1]. Saat ini penderita penyakit gagal ginjal kronis bukan hanya dialami oleh orang tua, akan tetapi dapat juga dialami oleh anak-anak dan remaja. Meningkatnya populasi penderita penyakit gagal ginjal ini disebabkan karena kurangnya kesadaran masyarakat akan gaya hidup sehat, contohnya seperti terlalu sering mengonsumsi makanan yang berlemak dan minum minuman bersoda, jarang berolahraga, bahkan merokok. Maka dari itu kesehatan ginjal manusia tergantung dari bagaimana mereka menjaga kesehatan ginjalnya. Pada tahun 2022 ini, terdapat banyak sekali kasus gagal ginjal yang meyerang anak-anak bahkan remaja. Kasus ini sangat hangat dibicarakan dikalangan masyarakat, karena penyebab dari penyakit ini disebabkan

oleh obat sirup yang terkandung etilen glikol (EG) dan dietilen glikol (DEG) yang melebihi ambang batas normal didalamnya.

Sebelumnya sudah banyak penelitian yang dilakukan untuk melakukan prediksi diantaranya menggunakan naïve bayes. Penelitian oleh Toni Arifin (2019), pada penelitian yang dilakukan adalah memprediksi penyakit gagal ginjal kronis menggunakan Algoritma Naïve Bayes Classifier berbasis Particle Swarm Optimization. Dari hasil penelitian Naive Bayes Classification berbasis Particle Swarm Optimization memiliki akurasi confusion matrix sebesar 98,75% dan AUC sebesar 99%. sedangkan Naive Bayes memiliki akurasi confusion matrix 97.00% dan AUC sebesar 99.8%[2]. Dalam penelitian ini akan dilakukan perbandingan dua metode yaitu metode Naïve Bayes dan metode KNN untuk memprediksi gagal ginjal kronis. Perbandingan dari dua metode ini adalah untuk mendapatkan model dengan ketinggian akurasi yang terbaik sehingga hasil prediksi bisa lebih akurat.

## II. STUDI PUSTAKA

### A. Machine Learning

Training data pada mesin untuk menghasilkan serangkaian input baru merupakan pengertian dari Machine learning. Tujuan utama dari Machine learning ini yaitu untuk memberikan suatu prediksi. Prediksi yang dimaksud merupakan keakuratan dari mesin yang di tes. Pengetesan pada Machine learning menggunakan sebuah set data yang diolah lagi menjadi set data pelatihan dan set data yang digunakan untuk memvalidasi pelatihan (validasi silang)[3]. Set data baru digunakan untuk membandingkan kedua jenis machine learning yang digunakan nantinya. Sebelum set data digunakan untuk memprediksi kedua machine learning, akan dibersihkan terlebih dahulu nilai yang tidak ada dan juga untuk klasifikasi yang class nya bukan angka akan di ubah terlebih dahulu.

### B. Pengklasifikasi Naïve Bayes

Naïve bayes classification adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class. Bayes classification didasarkan pada teorema Bayes yang memiliki kemampuan klasifikasi serupa dengan decision tree dan neural network. Bayesian classification terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam database dengan data yang besar. kuat. Oleh karena itu, Naive Bayes merupakan model yang memiliki fitur independen[4]. Naive Bayes digunakan untuk mendiagnosis dan memprediksi masalah di dunia. Untuk klasifikasi, algoritma naive bayes membutuhkan lebih sedikit data pelatihan untuk prediksi dan evaluasi parameter[5]. Metode klasifikasi Bayesian naif yang digunakan untuk memprediksi partner di setiap kelas. Misalnya, probabilitas untuk kumpulan data tertentu dari kelas target. Kelas dengan probabilitas tertinggi diharapkan menjadi kelas dengan probabilitas tertinggi. Aturan Klasifikasi Klasifikasi adalah proses menemukan model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep. Model diturunkan berdasarkan analisis kumpulan data pelatihan. Model ini digunakan untuk memprediksi penunjukan kelas objek dengan penunjukan kelas yang tidak diketahui[6]

Terdapat 2 persamaan teorema bayes yang nantinya sama- sama digunakan untuk menghitung probabilitas,  $P(c|x)$ , dari  $P(c)$ ,  $P(x)$ , dan  $P(x|c)$ . kelas kemerdekaan besyarat mengasumsi bahwa efek dari nilai a predictor (x) dengan kelas tertentu(c) tidak bergantung dengan nilai prediktor lainnya pada pengklasifikasian Naïve Bayes[7]

$$P(c|X) = \frac{P(X|C)P(c)}{P(x)} \quad (1)$$

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c) \quad (2)$$

$P(c|x)$  = probabilitas posterior kelas (target) diberikan prediktor (atribut).

$P(c)$  = probabilitas kelas sebelumnya.

$P(x|c)$  = peluang yang merupakan peluang kelas yang diberikan oleh prediktor.

$P(x)$  = probabilitas sebelumnya dari predictor.

### C. K-Nearest Neighbor algorithm (K-NN)

Algoritma K-Nearest Neighbors (K-NN) banyak digunakan untuk klasifikasi dan regresi dalam Pengenalan Pola Konsistensi data. Dalam keluarga algoritma K-NN, algoritma pembelajaran yang diawasi. K-NN adalah nonparametrik Sebuah metode, yang digunakan untuk mengevaluasi statistik. untuk keduanya Ketika , data input berasal dari blok pelatihan. Ada masukan pelatihan, tujuan dan hasil yang sesuai Model dihasilkan sebagai model. K-NN adalah tipenya Pembelajaran berbasis memori karena hipotesis dibangun secara langsung Dari contoh pelatihan Anda. tempat kelahiran tetangga Sekumpulan objek dari kelas yang diketahui. Untuk  $K = 1$ , Tetangga terdekat ditugaskan ke kelas. umumnya Skema penimbangan. garis lurus selalu dihasilkan Ada jarak terpendek antara dua tetangga, Jarak ini disebut jarak Euclidean[8]dengan K-NN Hasil regresi akan menjadi rata-rata dari  $k$  . itu Kelemahan dari algoritma K-NN tetangga terdekat adalah bahwa Bukan algoritma tercepat tetapi bekerja dengan input yang lebih sedikit. Fungsionalitas seragam diperlukan, sensitif lokal Penyelarasan data[9]

## III. METODOLOGI PENELITIAN

### A. Metode Penelitian

Pada penelitian ini peneliti menggunakan metode eksperimen. Dimana peneliti melakukan percobaan ini menggunakan dataset dari *UCI Machine learning*. Dataset ini digunakan untuk *training* data yang berfungsi sebagai pencari tingkat akurasi antara metode Naive Bayes dan metode K-Nearest Neighbor untuk menentukan prediksi gagal ginjal kronis dengan mengguakan bahasa pemograman Python dan untuk pengklisifikasiannya menggunakan library Sklearn. Peneliti akan menguji antara kedua sistem untuk dilihat tingkat keakuratan dengan cara membandingkan hasil akurasi dari kedua sistem.

### B. Pengumpulan Data

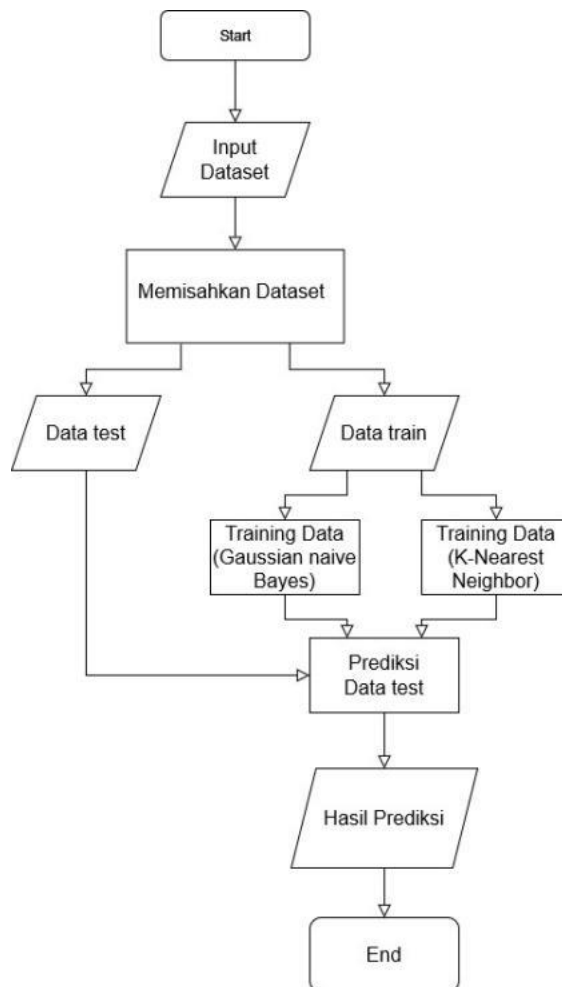
Dataset yang digunakan yaitu *chronic kidney disease* yang diambil dari internet dan sudah tersebar di publik. Dari dataset ini peneliti menggunakannya untuk melakukan *testing* dan *training* data yang berjumlah 281 data nyata dimana data tersebut terdapat 24 fitur, deskripsi, dan *values* yang ditampilkan pada tabel 1. untuk pengambiln dataset ini kami mencari menggunakan internet melalui laman [website:https://archive.ics.uci.edu/ml/datasets/Chronic\\_Kidney\\_Disease](https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease). [10]

TABLE I. FITUR DATASET

Sr.no	Attribute	Description	Values
1	age	age	Numerical
2	bp	Blood pressure	Numerical Mm/hg
3	sg	Specific gravity	nominal
4	al	albumin	nominal
5	su	sugar	nominal
6	rbc	Red blood cells	nominal
7	pc	Pus cell	nominal
8	pcc	Pus cell clumps	nominal
9	ba	bacteria	nominal
10	bgr	Blood glucose random	Numerical Mgs/dl
11	bu	Blood urea	Numerical Mgs/dl
12	sc	Serum creatinine	Numerical Mgs/dl
13	sod	sodium	Numerical mEq/L
14	pot	potassium	Numerical mEq/L
15	hemo	hemoglobin	Numerical gms
16	pcv	Packed cell voume	Numerical Cell/cumm

17	wc	White blood cell count	Numerical Cell/cumm
18	rc	Red blood cell count	Numerical Millions/cmm
19	htn	hypertension	1= yes 0 = no
20	dm	Diabetes melitus	1= yes 0 = no
21	cad	Coronary artery disease	1= yes 0 = no
22	appet	appetite	1= good 0 = poor
23	pe	Pedal edema	1= yes 0 = no
24	ane	anemia	1= yes 0 = no
25	class	class	1= ckd 0 = notckd

### C. Rancangan Algoritma Sistem



Gambar 1. Flowchart perbandingan sistem prediksi gagal ginjal

Pada penelitian kali ini peneliti telah merancang skema perbandingan sistem prediksi gagal ginjal seperti pada Gambar 1. dari penelitian ini kami memiliki tujuan untuk memprediksi gagal ginjal sehingga dapat membantu mendiagnosa pasien. Pada Gambar1, Flowchart dimulai dengan memasukkan *dataset* yang selanjutnya memisahkan dataset menjadi *data test* dan *data train*, kemudian *data train* di *training* menggunakan *Gaussian Naive Bayes* dan *K-Nearest Neighbor* dan terakhir dilakukan pengujian dari hasil training data dan data test dan didapatkanlah hasil dari prediksinya.

IV. HASIL DAN PEMBAHASAN

I. Naïve Bayes

A. Label Encoder

Dalam pemrosesan data, sebuah data harus berbentuk numerik, oleh karena itu data yang berbentuk nominal harus diubah terlebih dahulu ke dalam bentuk numerik. Sehingga data yang berbentuk nominal dapat di proses.

	sg	al	sc	hemo	pcv	htn	classification
0	1.020	1.0	1.2	15.4	28	1	0
1	1.020	4.0	0.8	11.3	22	0	0
2	1.010	2.0	1.8	9.6	15	0	0
3	1.005	4.0	3.8	11.2	16	1	0
4	1.010	2.0	1.4	11.6	19	0	0

Gambar 1. Label Encoder

B. Normalize Data

Tahapan ini adalah tahapan penting dalam pra proses Machine learning. *Normalize* data berfungsi membuat beberapa variabel data memiliki nilai yang sama, sehingga tidak ada data yang terlalu besar maupun terlalu kecil. Pada percobaan ini kami menggunakan metode *Min-Max*, Seperti yang terlihat pada Gambar 2.

Terdapat normalisasi data. Setiap data dikurang dengan data minimal, kemudian dibagi dengan selisih nilai maksimum dan minimum pada fitur tersebut. Sehingga *output* dari data ini memiliki range antara 0 hingga 1.

	sg	al	sc	hemo	pcv	htn
271	0.020101	0.0	0.001323	0.746835	0.674419	0.0
102	0.005025	0.0	0.022487	0.753165	0.906977	0.0
95	0.010050	0.2	0.009259	0.000000	0.139535	0.5
49	0.005025	0.4	0.034392	0.487342	0.372093	0.5
139	0.010050	0.4	0.031746	0.575949	0.465116	0.5
..	...	...	...	...	...	...
232	0.010050	0.2	0.021164	0.000000	0.139535	0.0
306	0.015075	0.0	0.010582	0.835443	0.906977	0.0
206	0.005025	0.2	0.046296	0.746835	0.651163	0.5
23	0.005025	0.0	0.021164	0.000000	0.139535	0.0
241	0.010050	0.2	0.014550	0.531646	0.418605	0.0

Gambar 2. Normalize data

C. Pemisahan Data

Data yang digunakan akan dipisah menjadi 2 data, data *training* dan data *test*. Pada Tabel 1. ditampilkan perbandingan persentase dan jumlah dari data *training* dan data *test*.

TABEL II. PERBANDINGAN DATA TRAINING DAN TEST

	Data Training	Data Test	Total
Persentase	80	20	100
Jumlah	320	80	400

### 1) Data Training

*Training* data adalah data yang akan digunakan untuk melatih program yang dibuat nantinya dapat membuat prediksi sehingga mampu mencari korelasi data sendiri atau belajar pola dari data yang diberikan. Berdasarkan pada Gambar 3. Data yang digunakan sebagai data latih sebanyak 80% dari jumlah total data keseluruhan yaitu 400 data sehingga data yang digunakan sebagai data latih berjumlah 320 data.

	sg	al	sc	hemo	pcv	htn
271	0.020101	0.0	0.001323	0.746835	0.674419	0.0
102	0.005025	0.0	0.022487	0.753165	0.906977	0.0
95	0.010050	0.2	0.009259	0.000000	0.139535	0.5
49	0.005025	0.4	0.034392	0.487342	0.372093	0.5
139	0.010050	0.4	0.031746	0.575949	0.465116	0.5
..	...	...	...	...	...	...
232	0.010050	0.2	0.021164	0.000000	0.139535	0.0
306	0.015075	0.0	0.010582	0.835443	0.906977	0.0
206	0.005025	0.2	0.046296	0.746835	0.651163	0.5
23	0.005025	0.0	0.021164	0.000000	0.139535	0.0
241	0.010050	0.2	0.014550	0.531646	0.418605	0.0

[320 rows x 6 columns]

Gambar 3. Data Training.

Data latih selanjutnya akan digunakan untuk melatih model dengan menggunakan Naive Bayes Metode Gaussian.

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train,y_train)
```

Gambar 4. Training Data Code

Pada Gambar 4 dilakukan penginputan Naive Bayes metode Gaussian yang berada di dalam modul sklearn. Setelah itu membuat variabel “classifier” sebagai pengganti fungsi GaussianNB(). Kemudian memasukkan data training ke dalam fungsi training “fit” untuk melatih model.

### 2) Data Test

Seperti pada Gambar 5, sisa 20% dari keseluruhan data digunakan sebagai data tes yaitu berjumlah 80 data. Data tes tidak boleh merupakan data yang pernah dilihat oleh model sebelumnya.

	sg	al	sc	hemo	pcv	htn
10	0.005025	0.4	0.047619	0.468354	0.348837	0.5
43	0.005025	0.2	0.137566	0.373418	0.255814	0.0
329	0.020101	0.0	0.006614	0.765823	0.813953	0.0
228	1.000000	0.4	0.027778	0.000000	0.139535	0.5
185	0.015075	0.2	0.002646	0.632911	0.488372	0.0
..	...	...	...	...	...	...
270	0.020101	0.0	0.009259	0.778481	0.651163	0.0
148	1.000000	0.4	0.630952	0.000000	0.139535	0.5
364	0.020101	0.0	0.003968	0.810127	0.744186	0.0
283	0.015075	0.0	0.021164	0.911392	0.697674	0.0
107	0.010050	0.2	0.031746	0.582278	0.488372	0.5

[80 rows x 6 columns]

Gambar 5. Data test

#### D. Data Prediksi

Data ini adalah data uji yang digunakan untuk melakukan klasifikasi berdasarkan dataset tes. Pada Gambar 6, total dari data prediksi sama dengan data tes yaitu 80 data. Hasil dari data prediksi dapat dilihat sebagai berikut:

```
y_pred = gnb.predict(X_test)
y_pred
array([0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1,
       0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1])
```

Gambar 6. Nilai Prediksi *Naive bayes*

#### E. Confusion Matrix

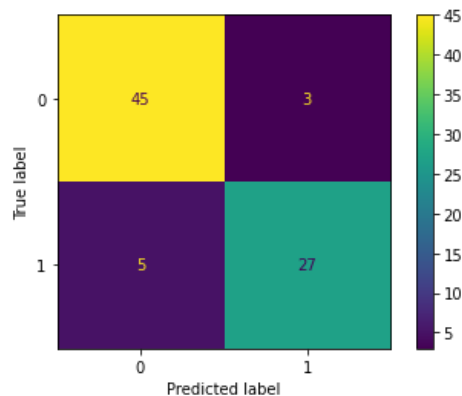
*Confusion Matrix* adalah sebuah tabel yang berisikan ringkasan perbandingan hasil prediksi dan hasil aktual. *Confusion Matrix* dapat digunakan untuk menilai kinerja model klasifikasi.

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay

cm = confusion_matrix(y_test, y_pred)*100
cm_display = ConfusionMatrixDisplay(cm).plot()
```

Gambar 7. *Confusion Matrix Code*

Pada Gambar 7 digunakan fungsi `confusion_matrix` dan `ConfusionMatrixDisplay` yang terdapat di dalam modul `sklearn`. Selanjutnya dilakukan pengecekan hasil data uji dengan “hasil yang sebenarnya” menggunakan fungsi `confusion_matrix`. Variabel “`cm_display`” digunakan untuk menggantikan fungsi `ConfusionMatrixDisplay().plot()` yang berfungsi untuk menampilkan hasil dari `confusion_matrix`.



Gambar 8. *Confusion Matrix*

Dari Gambar 8, matriks kebingungan menunjukkan bahwa 72 data yang terprediksi dinyatakan benar sesuai dengan “hasil yang sebenarnya”. Di antaranya sebanyak 45 data yang terprediksi benar tidak mengidap penyakit ginjal kronis dan data yang terprediksi benar mengidap penyakit ginjal kronis berjumlah 27 data. Kemudian terdapat 3 data terprediksi mengidap sakit jantung dan 5 data terprediksi tidak mengidap penyakit jantung adalah salah. Sehingga total data yang terprediksi salah berjumlah 8 data.

#### F. Tingkat Akurasi

Dari data yang diprediksi, data yang benar berjumlah 72 sedangkan data yang salah berjumlah 8 buah data. Sehingga nilai akurasi dari model ini menggunakan persamaan (3) adalah sebagai berikut:

$$\begin{aligned}
 \text{Akurasi} &= \frac{\Sigma \text{Prediksi Benar}}{\Sigma \text{Prediksi}} \times 100\% \\
 &= \frac{72}{80} \times 100\% \\
 &= 90\%
 \end{aligned}
 \tag{3}$$

Nilai ini sesuai dengan nilai yang terdapat pada program di bawah ini.

```

from keras.metrics.metrics import accuracy
from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test, y_pred)*100
print("Akurasi: {:.2f}%".format(acc))

Akurasi: 90.00%

```

Gambar 9. Nilai Akurasi Pada Python

Pada gambar 9. dilakukan penginputan perhitungan untuk menentukan nilai dari akurasi (*accuracy\_score*) yang terdapat didalam modul *sklearn*. Setelah itu dibuat variabel "acc" sebagai pengganti dari fungsi *accuracy\_score* yang digunakan untuk menghitung akurasi dari data yang telah diuji. Fungsi *print()* digunakan untuk menampilkan nilai akurasi dengan mengambil nilainya sampai ke 2 angka di belakang koma.

## II. K-Nearest Neighbours

### A. Data Prediksi

Data ini adalah data uji yang digunakan untuk melakukan klasifikasi berdasarkan dataset tes. Pada Gambar 10, total dari data prediksi sama dengan data tes yaitu 80 data. Hasil dari data prediksi dapat dilihat sebagai berikut:

```

[32] yhat = knn.predict(X_test)
     yhat[0:80]

array([0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1,
       0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0])

```

Gambar 10. Nilai Prediksi KNN

### B. Mencari Nilai Rata – Rata Akurasi

Nilai rata – rata akurasi digunakan untuk menentukan nilai K terbaik. Saat nilai rata rata mencapai nilai tertinggi dari situ nilai K dapat ditentukan

```

Ks = 10
mean_acc = np.zeros((Ks-1))
std_acc = np.zeros((Ks-1))
ConfustionMx = []
for n in range(1,Ks):

    #Train Model and Predict
    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train,y_train)
    yhat=neigh.predict(X_test)
    mean_acc[n-1] = metrics.accuracy_score(y_test, yhat)

    std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

mean_acc

array([0.95 , 0.9375, 0.9625, 0.925 , 0.925 , 0.925 , 0.925 , 0.925 ,
       0.925 ])

```

Gambar 11. Rata - Rata Akurasi

### C. Menentukan Nilai-K

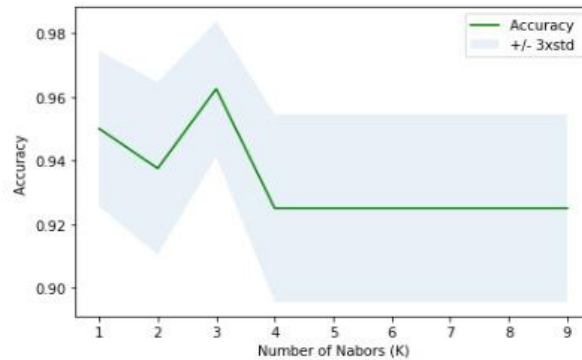
Nilai K dapat menentukan seberapa bagus tingkat suatu prediksi, oleh karena itu untuk menentukan nilai K yang baik digunakan sebuah metode untuk mencarinya,



```
plt.plot(range(1,Ks),mean_acc,'g')
plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc, alpha=0.10)
plt.legend(('Accuracy ', '+/- 3xstd'))
plt.ylabel('Accuracy ')
plt.xlabel('Number of Nabors (K)')
plt.tight_layout()
plt.show()
```

Gambar 12. Menampilkan Grafik K dan Mean Acc

Kode tersebut berguna untuk menampilkan rata – rata dan nilai K (Jumlah Nei pada sebuah grafik, module “plt”.



Gambar 13. Grafik Mean dan K

Terlihat pada grafik untuk nilai K = 3 nilai rata – rata akurasi tertinggi, oleh karena itu nilai K=3 akan digunakan untuk menentukan sistem prediksi metode KNN

#### D. Tingkat Akurasi

Tingkat akurasi pada KNN terlihat pada grafik yaitu sebesar 0,96 atau 96%

## V. KESIMPULAN DAN SARAN

Pada penelitian ini kami membuat sebuah alat bantu dalam bidang medis untuk memprediksi penyakit ginjal akut, kemudian kami membandingkan 2 metode klasifikasi yaitu Naïve Bayes dan juga KNN. Kami menemukan bahwa klasifikasi Naïve Bayes, dan KNN mempunyai tingkat prediksi yang sangat tinggi, masing-masing metode memiliki kelebihan dan kelemahan masing masing. Naïve Bayes memiliki tingkat keakuratan prediksi yang sedikit lebih kecil dibanding dengan tingkat keakuratan KNN. Metode Naïve Bayes lebih mudah dilakukan dibandingkan dengan metode KNN, pada metode KNN, dibutuhkan usaha lebih untuk mencari jumlah neighbours atau konstanta K. Berdasarkan penelitian yang telah dilakukan, penulis menyarankan untuk perbandingan antara dua metode dapat dicoba menggunakan data set lainnya dengan jumlah yang bervariasi sehingga dapat ditentukan metode mana yang lebih akurat untuk dipakai.

## DAFTAR PUSTAKA

- [1] A. N. Faddillah, J. Wijaya, and R. Hidayat, “Penerapan Algoritma Naive Bayes Untuk Diagnosa Penyakit Gagal Ginjal Kronis,” *J. Inf. ...*, vol. 18, no. 2, pp. 102–106, 2019, doi: 10.36054/jict-ikmi.v18i2.69.
- [2] T. Arifin and D. Ariesta, “Prediksi Penyakit Ginjal Kronis Menggunakan Algoritma Naive Bayes Classifier Berbasis Particle Swarm Optimization,” *J. Tekno Insentif*, vol. 13, no. 1, pp. 26–30, 2019, doi: 10.36787/jti.v13i1.97.
- [3] Dimsyar M Al Hafiz, Khoirul Amaly, Javen Jonathan, M Teranggono Rachmatullah, and Rosidi, “Sistem Prediksi Penyakit Jantung Menggunakan Metode Naive Bayes,” *J. Rekayasa Elektro Sriwij.*, vol. 2, no. 2, pp. 151–157, 2021, doi: 10.36706/jres.v2i2.29.
- [4] A. P. Wibawa *et al.*, “Naïve Bayes Classifier for Journal Quartile Classification,” *Int. J. Recent Contrib. from Eng. Sci. IT*, vol. 7, no. 2, p. 91, 2019, doi: 10.3991/ijes.v7i2.10659.
- [5] S. Danish, S. Aamer, S. P. Kharde, and S. S. Gadekar, “REVIEW ON CHRONIC KIDNEY DISEASE USING NAÏVE BAYES ALGORITHM,” no. 08, pp. 1520–1522, 2020.
- [6] 4. R. Agrawal and G. Psaila, “No Title,” *Act. data Min.*, 1995.
- [7] K. Vembandasamy, R. Sasipriya, and E. Deepa, “Heart Diseases Detection Using Naive Bayes Algorithm,” *Int. J. Innov. Sci. Eng. Technol.*, vol. 2, no. 9, pp. 441–444, 2015.
- [8] B. Venkata Ramana, M. S. P. Babu, and N. . Venkateswarlu, “A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis,” *Int. J. Database Manag. Syst.*, vol. 3, no. 2, pp. 101–114, 2011, doi: 10.5121/ijdms.2011.3207.
- [9] R. Devika, S. V. Avilala, and V. Subramaniaswamy, “Comparative study of classifier for chronic kidney disease prediction using

naive bayes, KNN and random forest," *Proc. 3rd Int. Conf. Comput. Methodol. Commun. ICCMC 2019*, no. Iccmc, pp. 679–684, 2019, doi: 10.1109/ICCMC.2019.8819654.

[10] "No Title." [https://archive.ics.uci.edu/ml/datasets/Chronic\\_Kidney\\_Disease](https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease).