Journal of Artificial Intelligence and Software Engineering
Vol. 5, No. 1, Maret 2025, pp. 176-187
E-ISSN: 2777-001X, DOI: 10.30811/jaise.v5i1.6456

Comparison of K-Nearest Neighbor and Naïve Bayes Algorithms for Tuberculosis Diagnosis Classification

Dedi Setiadi[1*], Alfis Arif[2], Anik Oktaria[3]
[1,2,3] Program Studi Teknik Informatika, Institut Teknologi Pagar Alam, Pagar Alam, 31520, Indonesia

Informasi Artikel

Diterima : 18 Februari 2025
Revisi : 27 Februari 2025
Publikasi : 20 Maret 2025

Kata Kunci:

Comparison
Tuberculosis
K-NN
Naïve Bayes
CRISP-DM

ABSTRAK

Tuberkulosis adalah penyakit menular yang disebabkan oleh bakteri *mycobacterium tuberculosis.* Tuberkulosis merupakan masalah kesehatan global yang serius dan dapat menyebabkan kematian bila tidak diobati dengan tepat. Di Puskesmas Sidorejo saat ini proses diagnosis pada pasien dengan menggunakan beberapa tolak ukur riwayat medis yang diperoleh dari pasien mengenai keluhan, gejala, dan faktor resiko, sedangkan perhitungan diagnosisnya belum diketahui hasilnya. Komparasi algoritma K-nearest neighbor dan naïve bayes dalam mengklasifikasi penyakit tuberkulosis, dapat memberi masukan untuk Puskesmas Sidorejo dalam melihat akurasi diagnosis penyakit tuberkulosis, dengan informasi medis seperti gejala dan riwayat kesehatan, dimana nantinya data pasien akan diolah menggunakan aplikasi *rapid miner*. Metode pengembangan sistem yang digunakan dalam penelitian ini ialah CRISP-DM, yang terdiri dari *business understanding, data understanding, data preparation, modeling, evaluation,* dan *deployment*. Metode pengujian menggunakan *confusion matrix* untuk mengukur akurasi model algoritma dengan hasilnya yaitu algoritma K-nearest neighbor menghasilkan akurasi yang tinggi sebesar 98 % sedangkan algoritma naïve bayes terendah dengan akurasi.70%.

ABSTRACT

Tuberculosis is an infectious disease caused by the bacteria mycobacterium tuberculosis. Tuberculosis is a serious global health problem and can cause death if not treated properly. At the Sidorejo Health Center, the current process of diagnosing patients uses several benchmarks of medical history obtained from patients regarding complaints, symptoms, and risk factors, while the results of the diagnosis calculation are not yet known. Comparison of the K-nearest neighbor and naïve bayes algorithms in classifying tuberculosis can provide input for the Sidorejo Health Center in seeing the accuracy of the diagnosis of tuberculosis, with medical information such as symptoms and medical history, where later patient data will be processed using the rapid miner application. The system development method used in this study is CRISP-DM, which consists of business understanding, data understanding, data preparation, modeling, evaluation, and deployment. The testing method uses a confusion matrix to measure the accuracy of the algorithm model with the results being that the K-nearest neighbor algorithm produces a high accuracy of 98% while the naïve bayes algorithm is the lowest with an accuracy of 70%.

Penulis Koresponden

Email: dedisetiadi1212@gmail.com

Cara sitasi IEEE::

D. Setiadi, A. Arif & A. Oktaria, "Comparison of K-Nearest Neighbor and Naïve Bayes Algorithms for Tuberculosis Diagnosis Classification" *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 1, 176-187, Maret 2025. doi: 10.30811/jaise.v5i1.6456

1. PENDAHULUAN

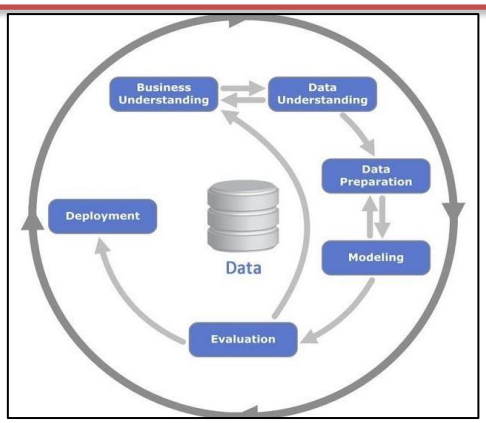
Tuberkulosis merupakan salah satu penyakit menular yang masih menjadi permasalahan kesehatan global, [1] terutama di negara berkembang seperti Indonesia. Berdasarkan data dari Organisasi Kesehatan Dunia (WHO), tuberkulosis menjadi salah satu penyakit menular penyebab utama kematian, [2] sehingga memerlukan diagnosis yang cepat serta akurat untuk menekan angka penyebarannya. Tuberkulosis adalah penyakit menular yang disebabkan oleh bakteri *mycobacterium tuberculosis*. [3] penyakit ini termasuk salah satu penyakit yang dicanangkan oleh *World Health Organization* (WHO) sebagai kedaruratan dunia *(global emergency)*, berdasarkan laporan WHO tahun 2009, indonesia termasuk negara terbesar ketiga setelah India dan Cina yang sebagian penduduknya terkena penyakit tuberkulosis, dan setiap tahunnya di indonesia terjadi sekitar 245.000 kasus tuberkulosis baru, dengan jumlah tuberkulosis menular (BTA+) sejumlah 107.000 kasus, sedangkan kematian karena tuberkulosis sekitar 46.000 setiap tahunnya. Tahun 2007 di Indonesia penyakit tuberkulosis merupakan salah satu dari sepuluh besar penyakit penyebab kematian terbesar di Indonesia. [4]

Jumlah penderita tuberkulosis yang terus meningkat sangat dipengaruhi oleh kondisi sosial-ekonomi masyarakat, [5] khususnya tingginya angka kemiskinan, pola hidup yang tidak sehat, serta lingkungan yang tidak bersih. Selain itu, kurangnya pemahaman tentang penyakit ini, termasuk gejala dan penyebabnya, turut memperburuk situasi. Tanpa informasi yang memadai, masyarakat akan kesulitan dalam mengenali tanda-tanda awal penyakit, yang menyebabkan keterlambatan dalam penanganan. Jika proses penanganan terlambat dan tidak dilakukan dengan tepat, kondisi pasien akan semakin memburuk, yang pada akhirnya bisa berujung pada komplikasi serius dan bahkan kematian. [6] Metode diagnosis tradisional, seperti uji *mikroskopis* dahak dan kultur bakteri, sering kali membutuhkan waktu yang lama serta bergantung pada ketersediaan fasilitas laboratorium yang memadai. Oleh karena itu, pengembangan sistem berbasis kecerdasan buatan *(artificial antelligence)* dapat menjadi solusi alternatif untuk meningkatkan efisiensi dan akurasi dalam mendeteksi tuberkulosis. [7]

Algoritma klasifikasi dalam *machine learning*, [8] telah banyak digunakan dalam berbagai bidang kehidupan manusia saat ini. Klasifikasi digunakan untuk membagi data dalam bebrbagai kategori atau kelas berdasarkan fitur yang ada, [9] adapun algoritma yang akan digunakan yaitu *K-Nearest Neighbors* dan *Naïve Bayes*, [10] termasuk dalam diagnosis penyakit tuberkulosis. *K-Nearest Neighbors* [11] dikenal sebagai algoritma berbasis kedekatan data yang sederhana namun efektif, [12] sementara *Naïve Bayes* memiliki keunggulan dalam kecepatan komputasi [13] dan kinerja yang baik pada data berskala besar. [14] Selain kelebihan dua model tersebut, terdapat juga kelemahannya yaitu model *K-Nearest Neighbors* lambat untuk *dataset* besar, KNN menghitung jarak setiap kali prediksi dilakukan, sehingga tidak efisien untuk dataset besar, sedangkan *naïve bayes* memiliki kelemahan yaitu kurang akurat untuk *dataset* kecil jika jumlah data latih terlalu sedikit, estimasi probabilitas bisa tidak stabil. [15] Namun, efektivitas kedua algoritma ini dalam klasifikasi diagnosis tuberkulosis masih perlu dikaji lebih lanjut, oleh sebab itu penelitian ini membandingkan performa algoritma *K-Nearest Neighbors* dan *Naïve Bayes* dalam mengklasifikasikan diagnosis penyakit tuberkulosis berdasarkan dataset medis yang tersedia. Perbandingan ini dilakukan berdasarkan beberapa parameter evaluasi seperti akurasi, presisi, *recall*, dan waktu eksekusi. Hasil dari penelitian ini yaitu dapat memberikan wawasan mengenai metode klasifikasi yang lebih optimal dalam membantu tenaga medis melakukan diagnosis tuberkulosis secara lebih cepat dan akurat.

2. METODE

Dalam penelitian ini, metode yang digunakan untuk proses klasifikasi diagnosis penyakit Tuberkulosis adalah CRISP-DM *(Cross-Industry Standard Process for Data Mining)*. CRISP-DM merupakan pendekatan standar dalam proses analisis data mining [16] yang terdiri dari enam tahapan utama, yaitu :



Gambar 1. Metode CRISP-DM

2.1. Business Understanding

Tahap ini bertujuan untuk memahami permasalahan yang ingin diselesaikan melalui data mining. [17] Dalam konteks penelitian ini, permasalahan yang diangkat adalah bagaimana membandingkan performa algoritma *K-Nearest Neighbors* dan *Naïve Bayes* dalam klasifikasi diagnosis tuberkulosis.

2.2. Data Understanding

Tahap ini melibatkan pengumpulan, eksplorasi, dan analisis awal dataset yang akan digunakan. Pada tahap ini, dilakukan pemeriksaan atribut dalam dataset, [18] distribusi data, dan identifikasi kemungkinan adanya data yang tidak lengkap atau tidak relevan. Data yang digunakan dalam penelitian ini adalah data yang diperoleh dari catatan rekam medis diagnosis di Puskesmas Sidorejo. Data yang diminta berupa gejala-gejala pada pasien, dan keseluruhan rekapan laporan diagnosis pada pasien . data tersebut akan diolah sehingga di dapat hasil hari akurasi terbaik dari algoritma *K-Nearest Neighbor* dan *Naïve Bayes*.

2.3. Data Preparation

Tahap ini mencakup proses pembersihan data, transformasi, serta pemilihan fitur yang relevan agar data siap digunakan dalam proses analisis. [19] Pada penelitian ini, proses normalisasi, penghapusan data duplikat, serta penanganan nilai yang hilang *(missing values)* akan dilakukan untuk meningkatkan kualitas dataset sebelum diterapkan ke model *machine learning*. Peneliti hanya menggunakan data yang memiliki atribut lengkap, agar selanjutnya data bisa dapat langsung diolah.

Tabel 1. Atribut penelitian

No	Atribut	Deskripsi	Hasil
1	Jenis kelamin	Jenis kelamin pasien	0 – Perempuan 1 – Laki-laki
2	Riwayat batuk	Riwayat batuk pasien	1 – Batuk-batuk 0 – Tidak batuk
3	Status merokok	Status merokok pasien	1 – Perokok 0 – Bukan perokok
4	Hasil rontegn	Hasil rontegn pasien	0 – Negatif 1 – Positif
5	Hasil pemeriksaan dahak	Hasil pemeriksaan dahak pasien	0 – Negatif 1 – Positif

2.4. Modeling

Setelah model dibangun, tahap evaluasi dilakukan untuk menilai performa masing-masing algoritma berdasarkan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. [20] Hasil evaluasi ini akan digunakan untuk menentukan algoritma mana yang lebih optimal dalam mengklasifikasikan diagnosis Tuberkulosis. Metode yang digunakan dalam penelitian ini ialah komparasi algoritma *K-Nearest Neighbor* dan *Naïve Bayes* sehingga didapat hasil dari pengujian *confusion matrix* dalam penelitian ini peneliti akan menggunkan *tools rapid miner*. Hasil pengujian dengan akurasi yang paling tinggi adalah algoritma yang akan dipakai untuk klasifikasi diagnosis penyakit tuberkulosis.

2.5. Evaluation

Setelah model dibangun, tahap evaluasi dilakukan untuk menilai performa masing-masing algoritma berdasarkan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Hasil evaluasi ini akan digunakan untuk menentukan algoritma mana yang lebih optimal dalam mengklasifikasikan diagnosis tuberkulosis. Pada tahap ini akan dilakukan evaluasi terhadap keefektifan dan kualitas model sebelum digunakaan dan menentukan apakah model dapat mencapai tujuan yang ditetapkan. [21] Pada komparasi algoritma *K-Nearest Neighbors* dan *Naïve Bayes* dengan tolak ukur yang digunakan adalah untuk mengukur tingkat akurasi dari algoritma yang digunakan. Pengujian yang digunakan oleh peneliti ialah *confusion matrix* yang akan digunakan pada pada aplikasi *rapid miner* yang pada umumnya untuk memberikan informasi perbandingan yang telah dilakukan oleh model yang digunakan dengan hasil klasifikasi sebenarnya untuk melihat akurasi tertinggi.

2.6. Deployment

Tahap terakhir dari CRISP-DM adalah penerapan model ke dalam sistem nyata. [22] Namun, dalam penelitian ini, tahap deployment akan berupa analisis hasil dan rekomendasi untuk pengembangan lebih lanjut agar model dapat digunakan dalam sistem pendukung keputusan medis. Pada tahap ini hasil dari model sudah keluar dan dapat digunakan dan mudah diketahui algoritma mana yang paling terbaik akurasinya. Dengan menggunakan metode CRISP-DM, penelitian ini dapat dilakukan secara sistematis, mulai dari pemahaman permasalahan hingga analisis hasil akhir. Pendekatan ini memastikan bahwa setiap tahapan dilakukan dengan baik sehingga menghasilkan model klasifikasi yang optimal dalam diagnosis penyakit tuberkulosis.

3. HASIL DAN PEMBAHASAN

3.1. Hasil

Penelitian ini menghasilkan model klasifikasi diagnosis penyakit tuberkulosis, dengan menggunakan algoritma *K-Nearest Neighbor* dan *Naïve Bayes*, yang menggambarkan performa akurasi. Data yang digunakan adalah data pasien tuberkulosis dari tahun 2019 – 2023 di Puskesmas Sidorejo. Hasil dari indikator – indikator tersebut merupakan rangkaian dari proses panjang dari algoritma *K-Nearest Neighbor* dan *Naïve Bayes*, mulai dari menentukan data transform, mencari nilai K kemudian menghitung jarak antara data testing dan data training. Mengurutkan jarak dari yang terkecil ke terbesar, lalu menentukan jumlah label mayoritas berdasarkan nilai k, selain itu juga engujian menggunakan *confusion matrix* untuk menghitung nilai akurasi. Dalam penelitian yang dilakukan dengan menggunakan *google colaboratory* untuk menentukan hasil akurasi dari algoritma *K-Nearest Neighbor* dan *Naïve Bayes*. Data total pasien yang digunakan selama 5 tahun dari tahun 2019 sampai dengan 2023 yaitu sebanyak 500 pasien.

3.2. Pembahasan

3.2.1. Model klasifikasi K-Nearst Neighbor

Permodelan klasifikasi *K-Nearst Neighbor* digunakan bahasa pemrograman *python* ada bebrapa tahapan yang harus dilakukan antara lain:

1. Library

Menginputkan library merupakan tahap pertama yang dilakukan karena sangat berpengaruh terhadap hasil program berikutnya . pada penelitian ini peneliti menggunakan library sebagai berikut

```
[ ] import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd
    import seaborn as sns
    from sklearn.metrics import mean_squared_error, r2_score, confusion_matrix
```

Gambar 2. *Library Python*

2. Menginput dan menampilkan data

Langkah selanjutnya yaitu menginputkan dataset yang akan digunakan, dengan menggunakan perintah seperti dibawah ini ;

```
] #Memasukkan Data
  df = pd.read_excel("datasett1.xls")
  df
```

Gambar 3. Menginput Dataset

Kemudian menampilkan data yang telah diinputkan. Seperti pada gambar dibawah ini :

	Usia	Jenis Kelamin	Riwayat Batuk	status merokok	Hasil Rotgen	hasil
0	30	1	1	1	1	0
1	6	1	0	0	1	1
2	50	1	1	0	1	0
3	48	0	0	0	1	1
4	50	1	0	0	1	1
...	...	...	...	...	...	...
496	29	1	0	0	0	1
497	56	1	1	0	1	1
498	30	1	0	0	0	1
499	76	0	1	1	1	0
500	56	0	1	0	1	0

501 rows × 6 columns

Gambar 4. Tampilan dataset

3. Melakukan pengecekan data

Data di cek untuk melihat adakah data yang kosong atau tidak dengan menggunakan perintah sebagai berikut ;

```
[ ]  #untuk menunjukkan apakah ada data yang kosong
     df.isnull().sum()
```

Gambar 5. Pengecekan Data

Setelah diinputkan kemudian akan tampil tampilan seperti dibawah ini :

```
Usia              0
Jenis Kelamin     0
Riwayat Batuk     0
status merokok    0
Hasil Rotgen      0
hasil             0
dtype: int64
```

Gambar 6. Tampilan Data

4. Menampilkan tiap jumlah klasifikasi pada atribut

Atribut yang dipakai ialah atribut hasil dengan mengubah variabelnya menjadi 0 dan 1 perintah yang diggunakan seperti dibawah ini ;

```
df['hasil'].value_counts()
```

Gambar 7. Memanggil Atribut

Kemudian akan tampil tampilan seperti dibawah ini :

```
hasil
1    350
0    151
Name: count, dtype: int64
```

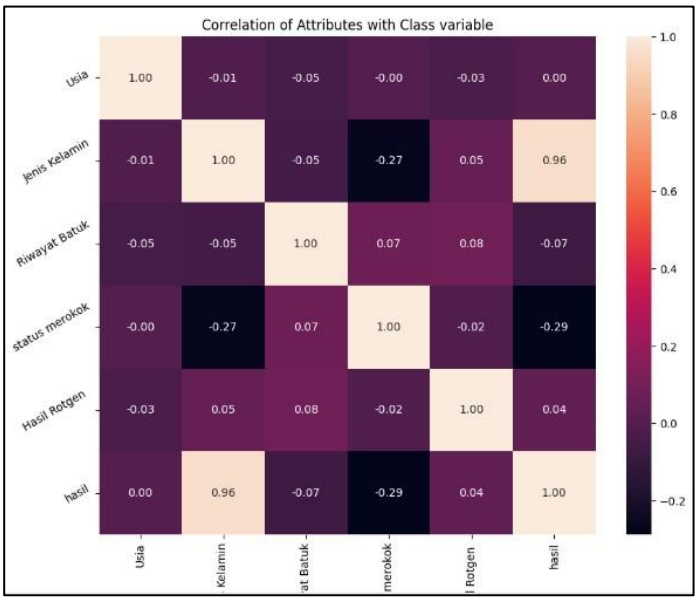
Gambar 8. Tampilan Atribut Hasil

Output dari atribut hasil tadi ialah menyatakan bahwa 1 = positif ialah sebanyak 350 pasien terdiagnosis tuberkolosis sedangkan 0 = negative sebanyak 151 pasien yang tidak terdiagnosis tb paru.

5. Menampilkan visualisasi korelasi data

```
#Menampilkan Visualisasi dari Korelasi data
plt.figure(figsize=(10,8))
plt.title('Correlation of Attributes with Class variable')
a = sns.heatmap(correlation, square=True, annot=True, fmt='.2f', linecolor='white')
a.set_xticklabels(a.get_xticklabels(), rotation=90)
a.set_yticklabels(a.get_yticklabels(), rotation=30)
plt.show()
```

Gambar 9. Koding menampilkan visualisasi



Gambar 10. Tampilan Visualisasi Korelasi

6. Mencari nilai k terbaik

Sebelum melakukan klasifikasi menggunakan algoritma KNN, Peneliti harus menentukan nilai K yang akan digunakan dengan menggunakan bantuan grid search yang ada di google colab untuk menentukan nilai K terbaik dari Dataset diatas.

```
# Nilai K terbaik
best_k = [np.argmax(k_range)]
print(f"Nilai K terbaik adalah: {best_k} dengan rata-rata akurasi: {max(k_range):.4f}")

Nilai K terbaik adalah: [29] dengan rata-rata akurasi: 30.0000
```

Gambar 11. Nilai K

Hasil dari pencarian K terbaik menggunakan bantuan google colab didapatkan jumlah k=29

7. Membagi data training dan testing

Setelah itu dataset digunakan untuk menentukan *hyperparameter* untuk data training dan data testing. Data testing diatur sebanyak 0.1 atau 10% dari total keseluruhan dataset dan random_state = 0 memiliki arti jika pemilihan data testing tidak akan berubah setiap kali kita mengatur nilainya dengan 0. Menunjukan jumlah baris dalam X_test.

Dengan menggunakan perintah seperti dibawah ini ;

```
#Melakukan pembagian data training dengan data testing, data training diambil 10% dari data yang digunankan
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 0)
X_train.shape, X_test.shape
```

Gambar 12. Pembagian Data

Maka akan menunjukan jumlah baris dalam X_test.Hasilnya ((450,5),(51,5) seperti tampilan dibawah ini ;

```
((450, 5), (51, 5))
```

Gambar 13. Hasil Pembagian Data

Langkah berikutnya menampilkan data training perintahnya sebagai berikut ;

```
#Menampilkan Data Training
X_train
```

Gambar 14. Menampilkan Data Training

Maka akan tampil output seperti dibawah ini ;

	Usia	Jenis Kelamin	Riwayat Batuk	status merokok	Hasil Rotgen
170	-0.331584	0.647506	0.647506	-0.778312	-1.376936
268	-0.977619	-1.544388	0.647506	1.284832	0.726250
144	-2.269691	-1.544388	0.647506	1.284832	0.726250
132	-0.285438	-1.544388	0.647506	-0.778312	0.726250
12	0.222161	0.647506	0.647506	1.284832	-1.376936
...	...	...	...	...	...
323	-1.069910	-1.544388	-1.544388	1.284832	0.726250
192	-0.793038	0.647506	-1.544388	-0.778312	-1.376936
117	1.237360	0.647506	0.647506	1.284832	0.726250
47	1.652669	0.647506	-1.544388	-0.778312	-1.376936
172	-0.285438	0.647506	0.647506	-0.778312	0.726250

450 rows × 5 columns

Gambar 15. Output Data Training

Selanjutnya menampilkan Data *Testing* dengan perintah dibawah ini;

```
#Menampilkan Data Testing
X_test
```

Gambar 16. Memanggil Data Testing

Maka akan tampil *output* seperti dibawah ini ;

	Usia	Jenis Kelamin	Riwayat Batuk	status merokok	Hasil Rotgen
90	1.052778	0.647506	0.647506	-0.778312	0.726250
254	0.360597	0.647506	0.647506	1.284832	0.726250
284	1.652669	0.647506	-1.544388	1.284832	-1.376936
446	1.237360	-1.544388	0.647506	1.284832	0.726250
339	-1.023765	0.647506	0.647506	1.284832	0.726250
15	-0.793038	-1.544388	-1.544388	-0.778312	-1.376936
407	-0.793038	0.647506	0.647506	-0.778312	0.726250
278	0.222161	0.647506	-1.544388	-0.778312	0.726250
159	0.637470	-1.544388	0.647506	-0.778312	0.726250
153	-0.331584	-1.544388	0.647506	1.284832	0.726250
241	0.222161	0.647506	-1.544388	-0.778312	-1.376936
250	-0.700747	-1.544388	-1.544388	1.284832	0.726250
306	-0.285438	0.647506	0.647506	-0.778312	0.726250
439	-1.346783	0.647506	0.647506	1.284832	-1.376936

Gambar 17. Output Data Testing

8. Memasukan algoritma K-Nearest Neighbors

Memasukan perintah algoritma knn ke dalam *google colab* dengan menggunakan perintah sebagai berikut ;

```
[15] #Memasukkan Algoritma KNN
     from sklearn.neighbors import KNeighborsClassifier
     knn = KNeighborsClassifier(n_neighbors=3,weights='distance',metric='euclidean')
     knn.fit(X_train, y_train)
```

Gambar 18. Memanggil Algoritma *K-Nearest Neighbors*

Setelah menginputkan perintah makan akan menampilkan tampilan sebagai berikut ;

```
KNeighborsClassifier
KNeighborsClassifier(metric='euclidean', n_neighbors=3, weights='distance')
```

Gambar 19. Tampilan Algoritma *K-Nearest Neighbors*

9. Menampilkan hasil klasifikasi algoritma K-Nearest Neighbors

Langkah berikutnya ialah memasukan perintah hasil klasifikasi dari algoritma knn dengan perintah sebagai berikut ;

```
#Menampilkan Hasil Klasifikasi dari algoritma KNN
y_pred = knn.predict(X_test)
y_pred
```

Gambar 20. Hasil Klasifikasi *K-Nearest Neighbors*

Setelah itu maka akan menampilkan keluaran sebagai berikut ;

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

Gambar 21. Output Klasifikasi *K-Nearest Neighbors*

3.2.2. Modeling klasifikasi Naïve Bayes

Permodelan klasifikasi *Naïve Bayes* digunakan bahasa pemrograman *python* ada bebrapa tahapan yang harus dilakukan antara lain :

1. Library

Menginputkan *library* merupakan tahap pertama yang dilkan karena sangat berpengaruh terhadap hasil program berikutnya . pada penelitian ini peneliti menggunakan library sebagai berikut :

```
[22] #Memasukkan Library
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
```

Gambar 22. *Library Python*

2. Menginput dan menampilkan data latih

Langkah selanjutnya yaitu menginputkan dataset yang akan digunakan, dengan menggunakan perintah seperti dibawah ini ;

```
#memasukan data latih
datalatih = pd.read_excel("datasett.xls")
datalatih.head(11)
```

Gambar 23. Memasukan Data Latih

Kemudian menampilkan data yang telah diinputkan. Seperti pada gambar 24 berikut:

	Usia	Jenis Kelamin	Riwayat Batuk	status merokok	Hasil Rotgen	hasil
0	89	1	1	0	1	1
1	67	1	0	0	1	0
2	89	1	1	0	1	1
3	56	0	1	0	1	0
4	78	1	1	1	1	0
5	45	1	1	1	1	1
6	34	1	1	0	0	1
7	98	0	0	1	0	1
8	45	1	1	0	1	0
9	23	1	0	1	1	1
10	56	1	0	1	1	1

Gambar 24. Tampilan data latih

3. Memvisualkan persebaran data

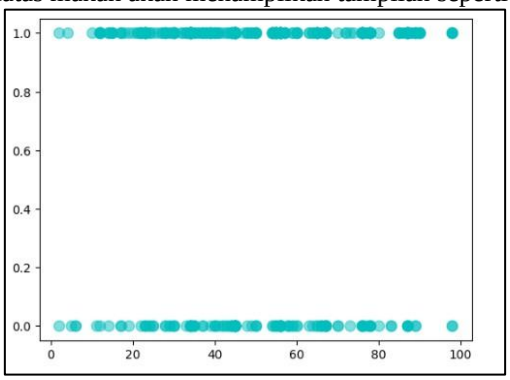
Langkah selanjutnya yaitu memvisualkan atau mengambarkan persebaran data dengan menggunakan perintah seperti dibawah ini;

```
#Memvisualkan persebaran data
from sklearn.cluster import KMeans

plt.scatter(datalatih.Usia, datalatih.hasil, s = 75, c = "c", marker = "o", alpha = 0.5)
plt.show()
```

Gambar 25. Memvisualkan Data

Setelah meinputkan perintah diatas makan akan menampilkan tampilan seperti pada gambar dibawah ini ;



Gambar 26. Hasil Visualisasi Data

4. Menentukan hasil probabilitas prediksi

Langkah berikutnya ialah menentukan hasil probabilitas dengan menggunakan sebagai berikut ;

```
#Menentukan probabilitas hasil prediksi
nbtrain.predict_proba(x_test)
```

Gambar 27. Menentukan Probabilitas

Setelah menentukan hasil probabilitas makan akan keluar tampilan sebagai berikut ;

```
array([[0.37861344, 0.62138656],
       [0.35305978, 0.64694022],
       [0.35071121, 0.64928879],
       ...,
       [0.35505479, 0.64494521],
       [0.19409373, 0.80590627],
       [0.19409373, 0.80590627]])
```

Gambar 28. Hasil Probabilitas

3.2.3. Hasil Akurasi Google Colab

Berdasarkan pengujian yang telah dilakukan dengan menggunakan *coufusion matrix* , supaya menghasilkan akurasi yang baik *fold coufusion matrix* dengan menggunakan *google colab* dengan jumlah keseluruhan data yang digunakan yaitu 500 data , Berikut ini adalah hasil akurasi perbandingan antara 2 algoritma yaitu *K-Nearest Neighbors* dan *Naïve Bayes* ;

1. Akurasi Algoritma K-Nearest Neighbor

Setelah melakukan pengujian model algoritma *K-Nearest Neighbor* menggunakan *google colab* dalam melakukan klasifikasi terhadap kelas dalam pengujian ini, barulah dapat dilihat hasil akurasi dari model algortima *Naïve Bayes*. Dalam penelitian ini, dengan data pasien dengan diagnosa *tuberculosis* yang telah dikumpulkan berhasil menunjukan akurasi sebesar 98%.

```
#Menampilkan Hasil akurasi Model
from sklearn.metrics import accuracy_score
print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, y_pred)))

Model accuracy score: 0.9804
```

Gambar 28. Memanggil Hasil Akurasi *K-Nearest Neighbors*

Dengan melihat hasil dari akurasi model *K-Nearest Neighbors* tersebut dapat digunakan untuk mengukur kinerja model seperti tingkat accuracy,precision, recall dan f-1 score dimana hasilnya sebagai berikut:

	precision	recall	f1-score	support
0	0.95	1.00	0.97	18
1	1.00	0.97	0.98	33
accuracy			0.98	51
macro avg	0.97	0.98	0.98	51
weighted avg	0.98	0.98	0.98	51

Gambar 29. Hasil Akurasi *K-Nearest Neighbors*

2. Akurasi Algoritma Naïve Bayes

Setelah melakukan pengujian model algoritma *Naïve Bayes* menggunakan *google colab* dalam melakukan klasifkasi terhadap kelas dalam pengujian ini barulah mengetahui akurasi dari algoritma *Naïve Bayes* dalam penelitian ini, dengan data oasien dengan diagnosa tuberkulosis sebanyak 500 data yang telah dikumpulkan berhasil menunjukan akurasi 70%

```
#Menampilkan hasil akurasi Naive Bayes
from sklearn.metrics import accuracy_score
accuracy= accuracy_score(y_uji, Y_predict)
print("Akurasi Naive Bayes : ",accuracy)

Akurasi Naive Bayes :  0.6986027944111777
```

Gambar 30. Memanggil Hasil Akurasi *Naïve Bayes*

Dengan melihat hasil dari akurasi model knn tersebut dapat digunakan untuk mengukur kinerja model seperti tingkat *accuracy,precision, recall* dan *f-1 score* dimana hasilnya sebagai berikut:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	151
1	0.70	1.00	0.82	350
accuracy			0.70	501
macro avg	0.35	0.50	0.41	501
weighted avg	0.49	0.70	0.57	501

Gambar 31. Hasil Akurasi Naïve Bayes

Dari hasil analisis *google colab tersebut*, bahwa algoritma *K-Nearest Neighbor* memiliki akurasi yang lebih tinggi dibandingkan dengan *Naïve Bayes* dalam klasifikasi penyakit tuberkulosis. Yaitu hasil dengan algoritma *K-Nearest Neighbor* mencapai akurasi 98%, sedangkan algoritma *Naïve Bayes* hanya mencapai akurasi 70%, yang terlihat pada table 4.

Tabel 2. Akurasi

Algoritma	Hasil Akurasi
K-Nearst Neighbor	98 %
Naïve Bayes	70 %

3. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan menggunakan algoritma *K-Nearest Neigbor* dan algoritma *Naïve Bayes* dengan menggunakan data pasien dengan diagnosa tuberkulosis di Puskesmas Sidorejo, dari komparasi kedua model algoritma tersebut diperoleh hasil yaitu nilai akurasi yaitu untuk algoritma K-nearest neigbor sebesar 98% sedangkan nilai akurasi algoritma naïve bayes yaitu sebesar 70 % dengan melalui perhitungan algoritma tersebut dengan menggunakan pemrograman python menggunakaan tools *google colabolatory*. Sehingga dapat dinyatakan bahwa algoritma *K-nearest Neigbor* lebih disarankan untuk diterapkan dalam mengklasifikasi diagnosa tuberkulosis di Puskesmas Sidorejo untuk mendapatkan hasil yang lebih akurat, karena hasil dari komparasi yang dilakukan dalam penelitian ini yaitu akurasi dari algoritma *K-Nearest Nighbor* lebih unggul dari algoritma *Naïve Bayes*.

REFERENSI

[1] R. Simamora, A. Alhafiz, and S. Julianita, "Sistem Pakar Mendiagnosis Tuberkulosis Pada Remaja Menggunakan Metode Dempster Shafer," *J. Sist. Inf. Triguna Dharma (JURSI TGD)*, vol. 3, no. 5, pp. 713–723, 2024.

[2] A. A. Prameswaty, M. H. P. Swari, and W. S. J. Saputra, "Perancangan Sistem Pakar Diagnosis Penyakit Tbc Paru Dengan Metode Certainty Factor Dan Dempster Shafer," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 5, pp. 8658–8663, 2024.

[3] A. A. Nabila, "Sistem Pakar Diagnosa Penyakit Tuberkulosis Dengan Metode Certainty," *J. Artif. Intell. Softw. Eng.*, vol. 3, no. 1, pp. 1–6, 2023.

[4] A. A. Alimi, A. R. Adriansyah, and P. Prima, "Pengembangan Sistem Deteksi Tuberkulosis pada Citra X-Ray Menggunakan Metode Convolutional Neural Network (CNN) dengan Framework Laravel," *J. Inform. Terpadu*, vol. 10, no. 2, pp. 165–171, 2024.

[5] M. R. Syahwana and R. M. Simanjorang, "Analisa Sistem Pakar Metode Bayes Dalam Mendiagnosa Penyakit Tubercolosis," *J. Sist. Informasi, Tek. Inform. dan Teknol. Pendidik.*, vol. 1, no. 2, pp. 57–66, 2022.

[6] D. S. Wulandari and M. G. Rohman, "Implementasi Metode Naïve Bayes Pada Sistem Pakar Diagnosa Penyakit Tuberculosis," *Gener. J.*, vol. 7, no. 3, pp. 64–76, 2023.

[7] M. Ula, A. Zulfikri, A. F. Ulva, and R. A. Rizal, "Penerapan Machine Learning Clustering K-Means dan Linear Regression Dalam Penentuan Tingkat Resiko Tuberkulosis Paru," *Indones. J. Comput. Sci.*, vol. 12, no. 1, 2023.

[8] W. Ramdhani, D. Bona, R. B. Musyaffa, and C. Rozikin, "Klasifikasi Penyakit Kangker Payudara Menggunakan Algoritma K-Nearest Neighbor," *J. Ilm. Wahana Pendidik.*, vol. 8, no. 12, pp. 445–452, 2022.

[9] A. Khaidar, M. Arhami, and M. Abdi, "Application of the Random Forest Method for UKT Classification at Politeknik Negeri Lhokseumawe," *J. Artif. Intell. Softw. Eng.*, vol. 4, no. 2, pp. 94–103, 2024.

[10] A. Ridwan, "Penerapan Algoritma Naïve Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus," *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 15–21, 2020.

[11] S. Widaningsih, "Penerapan Data Mining untuk Memprediksi Siswa Berprestasi dengan Menggunakan Algoritma K Nearest Neighbor," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 2598–2611, 2022.

[12] E. Novianto, A. Hermawan, and D. Avianto, "Klasifikasi Algoritma K-Nearest Neighbor, Naive Bayes, Decision Tree Untuk Prediksi Status Kelulusan Mahasiswa S1," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 8, no. 2, pp. 146–154, 2023.

[13] M. Z. Haq, C. S. Octiva, A. Ayuliana, U. W. Nuryanto, and D. Suryadi, "Algoritma Naïve Bayes untuk Mengidentifikasi Hoaks di Media Sosial," *J. Minfo Polgan*, vol. 13, no. 1, pp. 1079–1084, 2024.

[14] T. P. R. Sanjaya, A. Fauzi, and A. F. N. Masruriyah, "Analisis sentimen ulasan pada e-commerce shopee menggunakan algoritma naive bayes dan support vector machine," *INFOTECH J. Inform. Teknol.*, vol. 4, no. 1, pp. 16–26, 2023.

[15] H. Budiantoro and B. Hendrik, "Implementasi SVM dan KNN pada Sistem Penunjang Keputusan Kenaikan Pangkat Guru," *J. KomtekInfo*, pp. 380–389, 2024.

[16] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Comput. Sci.*, vol. 181, pp. 526–534, 2021.

[17] D. Ruswanti, D. Susilo, and R. Riani, "Implementasi Crisp-Dm Pada Data Mining Untuk Melakukan Prediksi Pendapatan Dengan Algoritma C. 45," *Go Infotech J. Ilm. Stmik Aub*, vol. 30, no. 1, pp. 111–121, 2024.

[18] S. Navisa, L. Hakim, and A. Nabilah, "Komparasi Algoritma Klasifikasi Genre Musik pada Spotify Menggunakan CRISP-DM," *J. Sist. Cerdas*, vol. 4, no. 2, pp. 114–125, 2021.
[19] D. Setiadi, S. Sasmita, and M. Yolanda, "Penerapan Algoritma Regresi Linier Berganda Untuk Memprediksi Hasil panen Padi Di Kota Pagar Alam," *Kesatria J. Penerapan Sist. Inf. (Komputer dan Manajemen)*, vol. 5, no. 2, pp. 337–438, 2024.
[20] D. Setiadi, S. Sasmita, and Y. I. Mukti, "Optimization Of Agricultural Production In South Sumatera Using Multiple Linear Regression Algorithm," *Knowbase Int. J. Knowl. Database*, vol. 4, no. 2, pp. 168–179, 2024.
[21] M. R. Muttaqin, T. I. Hermanto, and M. A. Sunandar, "Penerapan K-Means Clustering dan Cross-Industry Standard Process For Data Mining (CRISP-DM) untuk Mengelompokan Penjualan Kue," *Komputasi J. Ilm. Ilmu Komput. dan Mat.*, vol. 19, no. 1, pp. 38–53, 2022.
[22] M. A. Hasanah, S. Soim, and A. S. Handayani, "Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir," *J. Appl. Informatics Comput.*, vol. 5, no. 2, pp. 103–108, 2021.