

ANALISIS METODE ALGORITMA K-NEAREST NEIGHBOR (KNN) DAN NAIVE BAYES UNTUK KLASIFIKASI DIABETES MELLITUS

Muhardi Saputra¹⁾, Johannes Putra Sidabuke²⁾, Ryan Pangeranta Sinulingga³⁾, Reslina Br Tamba⁴⁾

¹⁻⁴Fakultas Sains Dan Teknologi, Universitas Prima Indonesia

email: muhardisaputra@unprimdn.ac.id, johannesputra2211@gmail.com,
pangerantar@gmail.com, reslinatamba85@gmail.com

Abstract



This study aims to compare the performance of two algorithms in detecting diabetes mellitus, which is a metabolic disorder caused by insufficient insulin production by the pancreas. In this research, we used two algorithms, namely Naive Bayes and K-Nearest Neighbor (KNN), to carry out analysis on the diabetes mellitus dataset used. The Naive Bayes algorithm is a statistical algorithm used to classify and predict the probability of certain classes. Meanwhile, the K-Nearest Neighbor (KNN) algorithm is used to classify new objects based on their similarity to nearby objects. This study utilized 9 variables, including number of pregnancies, glucose levels, blood pressure, skin thickness, insulin, Body Mass Index (BMI), family history of diabetes, age, and diagnosis results. The dataset used consists of 2000 data obtained from KAGGLE. The classification process is carried out by importing data into Microsoft Excel, designing the process, and then analyzing the data using Google Colab by applying the K-Nearest Neighbor and Naive Bayes algorithms. The research results show that the K-Nearest Neighbor algorithm provides a higher level of accuracy compared to the Naive Bayes algorithm.

Keywords: Klasifikasi, Diabetes Mellitus, K-Nearest Neighbor, Naive Bayes.

1. PENDAHULUAN

Saat ini Diabetes Mellitus atau disebut dengan kencing manis, merupakan permasalahan kesehatan serius bagi masyarakat di Indonesia. Diabetes Mellitus merupakan penyakit metabolik dengan tanda tingkat gula darah yang tinggi, karena gangguan produksi insulin, kerja insulin, atau keduanya. Penyakit ini berlangsung sepanjang hidup seseorang dan merupakan kondisi kronis. Hasil dari berbagai studi epidemiologi menunjukkan bahwa insiden dan prevalensi DM cenderung meningkat di berbagai negara, sehingga menjadikan Diabetes Mellitus sebagai masalah kesehatan global yang sedang meningkat [1].

Penyakit diabetes mellitus terus meningkat dari tahun ke tahun. Setiap tahun, satu juta orang menjadi korban penyakit ini. Diabetes mellitus dapat mengalami perbaikan jika pasien mendeteksinya dan menjalani pengobatan sejak awal sebelum keparahannya bertambah. Oleh karena itu, penelitian

dibutuhkan untuk mengembangkan klasifikasi penyakit diabetes mellitus [2].

Dalam penelitian ini, digunakan metode K-Nearest Neighbor dan Naive Bayes untuk melakukan proses klasifikasi. Pengklasifikasian penyakit diabetes Mellitus menggunakan metode K-Nearest Neighbor dan Naive Bayes akan menjadi fokus pengembangan di masa depan. Keunggulan dari metode K-Nearest Neighbor mencakup pelatihan yang sangat cepat, sederhana, dan efektif pada dataset yang besar [3]–[7]. Namun, terdapat beberapa kelemahan dalam metode K-Nearest Neighbor, seperti kebutuhan untuk menentukan nilai k, kompleksitas komputasi, keterbatasan memori, serta potensi pengaruh atribut yang tidak relevan [8].

Metode Naive Bayes adalah teknik klasifikasi yang memanfaatkan prinsip Probabilitas Bayes untuk mengestimasi kelas yang belum diketahui [9]–[16]. Dalam Naive Bayes, asumsi yang digunakan adalah bahwa

dataset dianggap memiliki independensi yang tinggi [17].

Pada penelitian sebelumnya dengan judul “Penerapan Algoritma *Naive Bayes* Untuk Klasifikasi Penyakit Diabetes Mellitus” menggunakan algoritma *naive bayes*, diperoleh hasil pengujian dengan algoritma *naive bayes* yang hanya mampu menghasilkan 80.60% [18].

Penelitian lainnya yang berjudul “Prediksi penyakit diabetes mellitus menggunakan metode C4.5” menggunakan algoritma C4.5. Berdasarkan pengujian yang dilakukan bahwa algoritma KNN hanya mampu menghasilkan 81,82% [19].

Dengan merujuk pada konteks sebelumnya, penelitian ini dilaksanakan dengan tujuan melakukan klasifikasi penyakit diabetes mellitus. Melalui penelitian ini akan dibandingkan hasil penggunaan algoritma KNN (*K-Nearest Neighbor*) dan *Naive Bayes* dalam proses klasifikasi penyakit diabetes mellitus.

2. METODE PENELITIAN

2.1 Jenis Penelitian

Penelitian ini merupakan penelitian kualitatif dimana penelitian ini bertujuan untuk memahami fenomena tentang apa yang dialami subjek, seperti *Pregnancies* (kehamilan), *Glucose*, *Blood Pressure* (tekanan darah), *Skin Thickness* (ketebalan kulit), *Insulin*, *BMI* (indeks massa tubuh), *Diabetes Pedigree Function* (fungsi keturunan diabetes), *Age* (usia), *Outcome*.

Penelitian ini menerapkan metode fenomenologi yang tujuannya adalah untuk memberikan gambaran mendalam atau deskriptif tentang pengalaman hidup. Pendekatan ini memungkinkan penelitian untuk memahami dan mengungkapkan pengalaman hidup dari perspektif individu yang menjadi subjek penelitian.

Dalam pendekatan fenomenologi, penulis berusaha untuk mendalami makna dari suatu peristiwa dan hubungannya dengan individu yang berada dalam konteks tertentu. Dalam penelitian ini, peneliti mengenali pengalaman yang dialami oleh pasien diabetes

melitus ketika menjalani perawatan luka akibat diabetes [20].

2.2 Penyakit Diabetes Mellitus

Diabetes Mellitus merujuk pada serangkaian gejala yang muncul ketika meningkatnya kadar glukosa darah seseorang sebagai akibat dari kurangnya hormon insulin secara keseluruhan dan dapat diukur. Insulin adalah salah satu hormon utama yang terlibat dalam mengatur kadar glukosa dan dihasilkan oleh sel beta di pankreas [21].

Menurut pedoman PERKENI, Diabetes Mellitus dapat didiagnosis jika seseorang memiliki kadar gula darah puasa melebihi 126 mg/dL atau jika hasil tes gula darah sewaktu menunjukkan angka di atas 200 mg/dL. Kadar gula darah dalam tubuh bervariasi sepanjang hari, cenderung meningkat setelah makan, dan biasanya kembali ke tingkat normal dalam waktu 2 jam [22].

Akibatnya, terjadi peningkatan kadar gula darah yang menyebabkan gangguan metabolisme karbohidrat, protein, dan lemak, yang dapat menghasilkan berbagai komplikasi kronis pada organ tubuh.

2.3 KNN (*K-Nearest Neighbor*)

K-Nearest Neighbor merupakan metode dalam data mining dan machine learning yang digunakan untuk melakukan klasifikasi dan regresi. Metode ini bekerja dengan cara mencari nilai terdekat dari suatu data point yang baru, berdasarkan jarak euclidean atau jarak lainnya dalam ruang fitur. Kemudian, berdasarkan mayoritas kelas dari nilai tersebut, data point yang baru akan diklasifikasikan ke salah satu kelas atau diberikan prediksi regresi.

2.4 Naive Bayes

Naive Bayes adalah suatu teknik klasifikasi yang berasal dari Teorema Bayes yang dikembangkan oleh ilmuwan Inggris, Thomas Bayes. Metode ini berfokus pada penggunaan probabilitas dan statistik untuk memproses informasi dari pengalaman masa lalu dengan tujuan memprediksi peristiwa di masa

depan. Salah satu karakteristik utama dari metode Naive Bayes adalah asumsi yang kuat tentang independensi antara setiap peristiwa.

Naive Bayes merupakan metode yang mencakup seluruh kelas keputusan dengan menghitung probabilitas bahwa kelas keputusan adalah benar, berdasarkan vektor informasi objek. Algoritma ini mengasumsikan bahwa atribut objek adalah independen. Probabilitas yang terlibat dalam menghasilkan prediksi akhir dihitung sebagai total frekuensi dari tabel keputusan yang disebut sebagai "master" [23].

3. HASIL DAN PEMBAHASAN

3.1 Analisis Data

Data yang digunakan untuk mengklasifikasi diabetes diambil dari situs data science resmi yaitu :

KAGGLE adalah platform komunitas yang populer di kalangan data scientist dan peneliti, yang menyediakan berbagai sumber data yang dapat digunakan dalam penelitian dan analisis data. Peneliti mengambil dataset diabetes dari KAGGLE pada tahun 2020, diperoleh 2000 data yang dapat diolah menggunakan *machine learning*.

3.2 Pengolahan Data

Selanjutnya dilakukan tahap pengolahan data yang bertujuan untuk mengubah data mentah menjadi informasi yang bermanfaat untuk proses pengambilan keputusan maupun penyusunan laporan. Adapun langkah-langkah yang dilakukan untuk mendapatkan nilai akurasi dalam mengklasifikasi diabetes mellitus, yakni mempersiapkan data, preprocessing data.

Pembentukan model, melakukan train dan test untuk mencari nilai akurasi dan prediksi yang akurat menggunakan *Machine Learning* dengan Metode Algoritma KNN dan naïve bayes. Pengolahan data pada penelitian ini dilakukan dengan menggunakan bahasa pemrograman Python melalui Google Colab.

3.2.1 Import Library

Sebelum mengolah data diperlukan mengimport library terlebih dahulu agar data

tersebut dapat divisualisasikan dalam bentuk diagram dan plot, juga *train and test* untuk mendapatkan nilai akurasi dan prediksi yang akurat. Dalam tahap ini peneliti menggunakan Google Colab untuk menjalankan *Machine Learning* untuk mengimport library seperti berikut.

```
[2] import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

import warnings
warnings.filterwarnings('ignore')
```

Gambar 1. Import Library

3.2.2 Menampilkan Dataset

Dataset dalam bentuk excel atau csv di upload melalui content, setelah di upload lakukan code untuk membaca data tersebut seperti berikut:

DATA KAGGLE

```
[7] df = pd.read_csv('/content/diabetes-dataset.csv')
df
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0
...
1995	2	75	64	24	55	29.7	0.370	33	0
1996	8	179	72	42	130	32.7	0.719	36	1
1997	6	85	78	0	0	31.2	0.382	42	0
1998	0	129	110	46	130	67.1	0.319	26	1
1999	2	81	72	15	76	30.1	0.547	25	0

2000 rows x 9 columns

Gambar 2. Dataset KAGGLE

3.2.3 Normalisasi Data, Mengubah Dataset Menjadi Nilai Rata-Rata

Pencarian nilai rata-rata dari dataset diperlukan untuk normalisasi data agar data tersusun dengan rapi supaya dapat mempermudah dalam mencari nilai MinMax pada dataset.

```
[45] col=['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']
for i in col:
    df[i].replace(0, df[i].mean(), inplace=True)
```

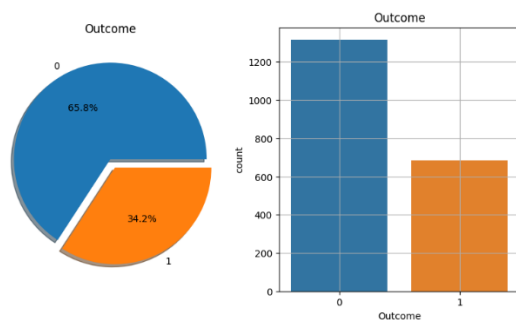
```
df.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	3.705500	121.970186	72.257047	25.932878	118.615412	32.643702	0.470930	33.095500	0.342000
std	3.306063	30.533180	11.968614	9.864003	88.366170	7.190337	0.323553	11.786423	0.474498
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	64.000000	20.955000	80.254000	27.600000	0.244000	24.000000	0.000000
50%	3.000000	118.000000	72.000000	23.000000	80.254000	32.300000	0.376000	29.000000	0.000000
75%	6.000000	141.000000	80.000000	32.000000	130.000000	36.800000	0.624000	40.000000	1.000000
max	17.000000	199.000000	122.000000	110.000000	744.000000	80.600000	2.420000	81.000000	1.000000

Gambar 3. Normalisasi Data

3.2.4 Visualisasi Outcome Count Plot

Diperlukan Outcome Plot untuk memvisualisasikan antara pasien yang terkena diabetes atau non diabetes agar mudah mengetahui persentase pada dataset tersebut.



Gambar 4. Grafik Presentase

3.2.5 Preprocessing Data

Preprocessing data adalah serangkaian langkah yang dilakukan untuk mempersiapkan data mentah sebelum dilakukan analisis atau pemodelan lebih lanjut. Tujuan preprocessing data adalah untuk mengoptimalkan kualitas data, memperbaiki kesalahan, menghilangkan noise, dan menjadikan data tersebut sesuai dengan teknik analisis yang akan digunakan.

```
[56] scaler = StandardScaler()
scaler.fit(x)
SSX = scaler.transform(x)
```

```
[57] x_train, x_test, y_train, y_test = train_test_split(SSX, y, test_size = 0.2, random_state = 7)
```

```
[58] x_train.shape, y_train.shape
```

```
((1600, 8), (1600,))
```

```
[59] x_test.shape, y_test.shape
```

```
((400, 8), (400,))
```

Gambar 5. Train and test

StandardScaler digunakan dalam preprocessing data untuk menormalkan atau menstandarisasi fitur numerik dalam dataset. Berikut adalah beberapa kegunaan dari StandardScaler:

1. Normalisasi data: StandardScaler dapat digunakan untuk mengubah skala fitur numerik dalam dataset sehingga memiliki mean 0 dan deviasi standar 1.
2. Menghilangkan outlier: StandardScaler dapat membantu mengurangi efek outlier dalam dataset.

3.2.6 Train dan test Dataset dengan metode Algoritma K-Nearest Neighbors

KNN

```
[60] from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
knn = KNeighborsClassifier(n_neighbors=9)
knn.fit(x_train, y_train)

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

print("Train Set Accuracy:" + str(accuracy_score(y_train, knn.predict(x_train))*100))
print("Test Set Accuracy:" + str(accuracy_score(y_test, knn.predict(x_test))*100))
```

Train Set Accuracy:83.875
Test Set Accuracy:88.0

Gambar 6. Metode KNN

Adapun kode yang digunakan untuk mengimplementasikan algoritma k-Nearest Neighbors (k-NN) menggunakan library scikit-learn. Kode ini melibatkan beberapa fungsi dan metode yang memiliki kegunaan berikut:

1. from sklearn.neighbors import KNeighborsClassifier: Ini mengimpor kelas KNeighborsClassifier dari modul neighbors dalam library scikit-learn.
2. From sklearn.metrics import accuracy_score, confusion_matrix, classification_report: codingan Ini mengimpor fungsi accuracy_score, confusion_matrix, dan classification_report dari modul metrics dalam library scikit-learn.
3. knn=KNeighborsClassifier(n_neighbors =9): Membuat objek klasifikasi knn dengan menggunakan algoritma k-NN. Parameter n_neighbors=9 menentukan

jumlah tetangga terdekat yang akan digunakan dalam klasifikasi.

4. `knn.fit (x_train,y_train)`: digunakan untuk melatih model K-NN.

Dengan menghitung dan mencetak akurasi pada data training dan testing, kita dapat hasil akurasi set yaitu train set accuracy 83,875 sedangkan test set accuracy 80,0.

3.2.7 Predict Dataset dengan menggunakan Metode Naïve bayes.

Untuk melakukan prediksi dengan menggunakan model Naive Bayes yang telah dilatih, Anda dapat menggunakan metode predict pada objek model yang telah dibuat. Berikut adalah contoh kode untuk melakukan prediksi menggunakan model Naive Bayes:

```
NAIVE BAYES

[27] from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
gnb.fit(x_train,y_train)

gnb_acc = accuracy_score(y_test,gnb.predict(x_test))

print("Train Set Accuracy:"+str(accuracy_score(y_train,gnb.predict(x_train))*100))
print("Test Set Accuracy:"+str(accuracy_score(y_test,gnb.predict(x_test))*100))

Train Set Accuracy:74.4375
Test Set Accuracy:79.75
```

Gambar 7. Metode Naive Bayes

1. `from sklearn.naive_bayes import GaussianNB`: Ini mengimpor kelas GaussianNB dari modul naive_bayes dalam library scikit-learn.
2. `gnb = GaussianNB()`: Membuat objek klasifikasi gnb dengan menggunakan algoritma Naive Bayes dengan distribusi Gaussian.
3. `gnb.fit (x_train, y_train)`: digunakan untuk melatih model Naive Bayes.

Dengan menggunakan metode naïve bayes dihasilkan train set accuracy 74,4735 dan test set accuracy 79,75.

3.2.8 Visualisasi grafik hasil train and test menggunakan confusion matrix

```
[45] from sklearn.metrics import confusion_matrix
labels = ['Class 0', 'Class 1']

# Hitung matriks kebingungan pada data testing
cm = confusion_matrix(y_train, knn.predict(x_train))

# Visualisasikan matriks kebingungan
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```



Gambar 8. Confusion Matrix

Melalui Confusion matrix (matriks kebingungan) pada gambar 8 diperoleh gambaran performa model klasifikasi dengan perbandingan prediksi yang dilakukan oleh model dengan label yang sebenarnya pada dataset testing.

3.4 Hasil

Setelah melakukan pembahasan sebelumnya, maka penulis mendapatkan hasil dari KNN Dan Naive Bayes Sebagai Berikut:

```
f,ax=plt.subplots(1,2,figsize=(10,5))
df['Outcome'].value_counts().plot.pie(explode=[0,0.1],autopct='%1.1f%%',ax=ax[0],shadow=True)
ax[0].set_title('Outcome')
ax[0].set_ylabel('')
ax[1].set_title('Outcome')
sns.countplot(data=df, x='Outcome')
N,P = df['Outcome'].value_counts()
print("Negative (0): ",N)
print("Positive (1): ",P)
plt.grid()
plt.show()

Negative (0): 1316
Positive (1): 684
```

Gambar 9. Hasil dari KNN dan Naive Bayes

3.4.1 Hasil K-Nearest Neighbor

Hasil akurasi K- Nearest Neighbor yaitu train set accuracy 84% sedangkan test set accuracy 80%.

3.4.2 Hasil Naive Bayes

Hasil akurasi Naive Bayes yaitu train set accuracy 74% dan test set accuracy 80%.

4. KESIMPULAN

Dari hasil temuan penelitian ini, dapat ditarik kesimpulan bahwa bahwa K-Nearest Neighbor dan metode Naïve Bayes dapat digunakan untuk mengklasifikasikan penyakit diabetes mellitus. Untuk pelatihan dan pengujian model, dataset yang digunakan terdiri dari 2000 data. Hasil pengujian menunjukkan bahwa model menggunakan algoritma K-Nearest Neighbor mencapai akurasi pelatihan sebesar 84% dan akurasi pengujian sebesar 80%. Sebaliknya, penggunaan metode Naïve Bayes hanya menghasilkan akurasi pelatihan sebesar 74% dan akurasi pengujian sebesar 80%. Selain itu, pengujian juga mengungkapkan bahwa matriks kebingungan menunjukkan tingkat true positive yang tinggi, tetapi tingkat true negative yang rendah. Dengan hasil ini, dapat disimpulkan bahwa metode K-Nearest Neighbor memiliki kemampuan yang lebih baik, dengan akurasi sebesar 84%, dalam mengklasifikasikan diabetes mellitus dengan efektif.

5. REFERENSI

- [1] D. O. Surya, R. Desnita, P. Studi, D. Keperawatan, and S. M. Padang, "EFEKTIFITAS INTERVENSI PEMBERDAYAAN KELUARGA DENGAN MODEL HOME CARE SERVICE TERHADAP MANAJEMEN DIRI PASIEN DIABETES MELITUS TIPE 2," vol. 12, no. 7, pp. 70–74, 2023.
- [2] D. W. Hestiana, "FAKTOR-FAKTOR YANG BERHUBUNGAN DENGAN KEPATUHAN DALAM PENGELOLAAN DIET PADA PASIEN RAWAT JALAN DIABETES MELLITUS TIPE 2 DI KOTA SEMARANG," *J. Heal. Educ.*, vol. 42, no. 3, pp. 138–145, 2017.
- [3] W. Aser, H. Samosir, and T. Gantini, "Analisis Dataset COVID-19 menggunakan Algoritma KNN dan Random Forest," *J. Strateg. - J. Maranatha*, vol. 4, no. 1, pp. 58–69, 2022, [Online]. Available: <https://mail.strategi.it.maranatha.edu/index.php/strategi/article/view/325>
- [4] S. A. Naufal, A. Adiwijaya, and W. Astuti, "Analisis Perbandingan Klasifikasi Support Vector Machine (SVM) dan K-Nearest Neighbors (KNN) untuk Deteksi Kanker dengan Data Microarray," *JURIKOM (Jurnal Ris. Komputer)*, vol. 7, no. 1, p. 162, 2020, doi: 10.30865/jurikom.v7i1.2014.
- [5] Y. Chen *et al.*, "Fast density peak clustering for large scale data based on kNN," *Knowledge-Based Syst.*, vol. 187, p. 104824, 2020, doi: 10.1016/j.knosys.2019.06.032.
- [6] N. A. Sinaga, K. Dalimunthe, and M. Sayid, "Komparasi Metode Decision Tree , KNN , dan SVM Untuk," *J. Sist. Komput. dan Inform.*, vol. 3, no. 2, pp. 94–100, 2021, doi: 10.30865/json.v3i2.3598.
- [7] L. andiani, J. Ilmu Komputer, and D. Palupi Rini, "Analisis Penyakit Jantung Menggunakan Metode KNN Dan Random Forest," *Pros. Annu. Res. Semin.*, vol. 5, no. 1, pp. 978–979, 2019.
- [8] S. Mutrofin, A. Izzah, A. Kurniawardhani, and M. Masrur, "OPTIMASI TEKNIK KLASIFIKASI MODIFIED K NEAREST NEIGHBOR MENGGUNAKAN ALGORITMA GENETIKA," *J. GAMMA*, no. September, pp. 130–134, 2014.
- [9] H. Mustofa and A. A. Mahfudh, "Klasifikasi Berita Hoax Dengan Menggunakan Metode Naive Bayes," *Walisongo J. Inf. Technol.*, vol. 1, no. 1, p. 1, Nov. 2019, doi: 10.21580/wjit.2019.1.1.3915.
- [10] J. Wu, Z. Cai, S. Zeng, and X. Zhu, "Artificial immune system for attribute weighted Naive Bayes classification," *Proc. Int. Jt. Conf. Neural Networks*, no. 61075063, 2013, doi: 10.1109/IJCNN.2013.6706818.
- [11] H. T. SIHOTANG, E. Panggabean, and H. Zebua, "Sistem Pakar Mendiagnosa

- Penyakit Herpes Zoster Dengan Menggunakan Metode Teorema Bayes,” vol. 3, no. 3, pp. 192–196, 2019, doi: 10.31227/osf.io/rjqgz.
- [12] A. M. Rahat, A. Kahir, and A. K. M. Masum, “Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset,” *Proc. 2019 8th Int. Conf. Syst. Model. Adv. Res. Trends, SMART 2019*, pp. 266–270, 2020, doi: 10.1109/SMART46866.2019.9117512.
- [13] H. Brawijaya, S. Samudi, and S. Widodo, “Komparasi Algoritma K-Nearest Neighbor dan Naive Bayes Pada Pengobatan Penyakit Kutil Menggunakan Cryotherapy,” *JUITA J. Inform.*, vol. 7, no. 2, pp. 93–100, 2019, doi: 10.30595/juita.v7i2.5609.
- [14] A. A. Murtopo, “Prediksi Kelulusan Tepat Waktu Mahasiswa STMIK YMI Tegal Menggunakan Algoritma Naive Bayes,” *CSRID (Computer Sci. Res. Its Dev. Journal)*, vol. 7, no. 3, p. 145, Jan. 2016, doi: 10.22303/csrid.7.3.2015.145-154.
- [15] A. Riani, Y. Susianto, and N. Rahman, “Implementasi Data Mining Untuk Memprediksi Penyakit Jantung Menggunakan Metode Naive Bayes,” *J. Innov. Inf. Technol. Appl.*, vol. 1, no. 01, pp. 25–34, 2019, doi: 10.35970/jinita.v1i01.64.
- [16] M. Dhea Noranita, Putri and Tri Basuki, Kurniawan and Edi, Suryanegara and Yesi Novaria, Kunang and Misinem, “Classification of MTI Student Thesis Documents at Bina Darma University Palembang Using Naive Bayes,” 2022.
- [17] I. L. F. Amien, W. Astuti, and K. M. Lhaksamana, “Perbandingan Metode Naive Bayes dan KNN (K-Nearest Neighbor) dalam Klasifikasi Penyakit Diabetes,” *e-Proceeding Eng.*, vol. 10, no. 2, pp. 1911–1920, 2023.
- [18] A. Ridwan, “Penerapan Algoritma Naive Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus,” *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 15–21, Oct. 2020, doi: 10.47970/siskom-kb.v4i1.169.
- [19] R. A. Siallagan and Fitriyani, “Prediksi Penyakit Diabetes Mellitus Menggunakan Algoritma C4.5,” *J. Responsif Ris. Sains dan Inform.*, vol. 3, no. 1, pp. 44–52, 2021, doi: 10.51977/jti.v3i1.407.
- [20] M. A. Prof. DR. Lexy J. Moleong, *Metodologi penelitian kualitatif*. Bandung: PT Remaja Rosdakarya, 2018.
- [21] E. Banjarnahor and S. Wangko, “Sel Beta Pankreas Sintesis Dan Sekresi Insulin,” *J. Biomedik*, vol. 4, no. 3, 2013, doi: 10.35790/jbm.4.3.2012.795.
- [22] N. S. Norma Lalla and J. Rumatiga, “Ketidakstabilan Kadar Glukosa Darah Pada Pasien Diabetes Melitus Tipe II,” *J. Ilm. Kesehat. Sandi Husada*, vol. 11, pp. 473–479, 2022, doi: 10.35816/jiskh.v11i2.816.
- [23] D. L. Olson and D. Delen, *Advanced data mining techniques*, no. December 2013. 2008. doi: 10.1007/978-3-540-76917-0.