Indonesian Journal of Computer Science
ISSN 2302-4364 (*print*) dan 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam, No. 1, Padang, Indonesia, Telp. (0751) 7056199, 7058325
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Komparasi Algoritma *K-Nearest Neighbors* dan *Naïve Bayes* dalam Klasifikasi Penyakit Diabetes Gestasional

Annisa Khoirala Ermy Pily[1], Oktavianda[2], Fanesa Aprilia[3], Rahmaddeni[4], Lusiana Efrizoni[5]
annkhoiermy210@gmail.com, oktavianda471@gmail.com, fanesaaprilia140403@gmail.com, rahmaddeni@sar.ac.id, lusiana@stmik-amik-riau.ac.id
STMIK Amik Riau

Informasi Artikel

Diterima : 23 Jan 2024
Direview : 24 Feb 2024
Disetujui : 27 Feb 2024

Kata Kunci

Diabetes Gestasional, *Feature Selection*, *K-Nearest Neighbors*, Klasifikasi, *Naïve Bayes.*

Abstrak

Diabetes merupakan penyakit metabolik dengan gejala hiperglikemia akibat gangguan sekresi insulin dan aksi insulin. Diabetes gestasional adalah gangguan toleransi glukosa pada wanita hamil. Saat kehamilan, plasenta menghasilkan hormon baru seperti human placental lactogen (HPL), hormon estrogen, dan hormon peningkat resistensi insulin. Gejala diabetes gestasional tidak selalu mudah dikenali, dan seringkali penderitanya mengalami gejala awal secara tidak sadar. Penelitian ini bertujuan untuk membandingkan performa dua algoritma yaitu *K-NN* dan *Naïve Bayes* dengan *Feature Selection* dalam mengklasifikasikan penderita diabetes gestasional. Hasil error terendah dari *feature selection* dengan iterasi K=4, memperoleh MAE 0.317, MSE 0.142, dan RMSE 0.377. Hasil akurasi pada model KNN dengan K=5 , tanpa *Feature Selection* sebesar 80% dan K-NN dengan *Feature Selection* sebesar 77%. Sementara itu, *Naïve Bayes* tanpa *Feature Selection* sebesar 77% dan *Naïve Bayes* dengan *Feature Selection* sebesar 80%. Dari hasil tersebut K-NN tanpa *Feature Selection* dan *Naïve Bayes* dengan *Feature Selection* mendapatkan hasil yang lebih baik.

Keywords

Calssification, Feature Selection, Gestational Diabetes, K-Nearest Neighbors, Naïve Bayes.

Abstrak

Diabetes is a metabolic disease with symptoms of hyperglycemia due to impaired insulin secretion and insulin action. Gestational diabetes is impaired glucose tolerance in pregnant women. During pregnancy, the placenta produces new hormones such as human placental lactogen (HPL), the hormone estrogen, and a hormone that increases insulin resistance. The symptoms of gestational diabetes are not always easy to recognize, and sufferers often experience the initial symptoms without realizing it. This study aims to compare the performance of two algorithms, namely K-NN and Naïve Bayes with Feature Selection in classifying gestational diabetes. The lowest error results from Feature Selection with K=4, obtained MAE 0.317, MSE 0.142, and RMSE 0.377. The accuracy results for K-NN with K=5, were 80% and K-NN with Feature Selection was 77%. Meanwhile, Naïve Bayes was 77% and Naïve Bayes with Feature Selection was 80%. Remarkably, K-NN and Naïve Bayes with Feature Selection get better results.

Attribution-ShareAlike 4.0 International License

A. Pendahuluan

Menurut International Diabetes Federation (IDF), diabetes adalah kondisi jangka panjang yang ditandai dengan kadar gula darah tinggi dalam tubuh. Jika kadar gula darah diabetes tidak terkontrol dengan baik, dapat menyebabkan berbagai komplikasi serius yang memengaruhi berbagai organ tubuh, seperti masalah mata, ginjal, dan saraf, serta meningkatkan risiko serangan jantung dan stroke [1]. Faktor genetik, serta perilaku atau gaya hidup yang tidak sehat, adalah penyebab utama diabetes. Selain itu, Faktor lingkungan sosial dan ketersediaan layanan kesehatan juga dapat memengaruhi timbulnya penyakit diabetes [2]. Ada beberapa tipe diabetes mellitus diantaranya: Diabetes tipe 1. (Bergantung pada pemakaian insulin). Diabetes tipe 2 (tidak bergantung pada insulin), Diabetes tipe lainnya dan diabetes gestasional (diabetes terhadap ibu hamil)[3].

Diabetes melitus gestasional (DMG) adalah gangguan toleransi glukosa yang pertama kali ditemukan pada wanita hamil. Ini terjadi pada wanita yang sebelumnya belum pernah didiagnosis diabetes dan kemudian menunjukkan kadar glukosa tinggi selama kehamilan [4]. Penyebabnya adalah pergantian hormon. Plasenta akan menghasilkan hormon baru, seperti human placental lactogen (HPL), hormon estrogen, dan hormon yang meningkatkan resistensi insulin selama kehamilan. Hormon-hormon ini juga akan meningkat selama kehamilan, yang juga mempengaruhi kerja insulin. Saat umur kehamilan mencapai 24 minggu, kondisi ini biasanya hilang dan kadar gula darah penderita kembali normal. Oleh karena itu, penanganan yang tepat sangat penting. Teknik data mining dapat digunakan dalam proses diagnosis [3]. Data mining adalah proses analisis yang bertujuan untuk menemukan pengetahuan dalam suatu basis data, seperti pola data atau hubungan antar data, khususnya dalam konteks pengolahan data berskala besar. Oleh karena itu, data mining memiliki peran yang sangat penting di berbagai bidang [5]. Untuk memastikan jumlah gula dalam tubuh, pengujian harus dilakukan berulang kali. Beberapa tanda kesehatan dapat menunjukkan bahwa tubuh memiliki penyakit diabetes. Namun, ada keterbatasan tenaga medis dan teknologi untuk mendeteksi secara manual data yang sangat besar. Jadi, peranan teknologi di bidang perhitungan matematika sangat membantu dalam bidang medis [6]. Dengan itu, teknik algoritma yang digunakan adalah classification atau klasifikasi. Metode yang akan digunakan dalam penelitian ini yaitu *K-Nearest Neighbors* (K-NN) dan *Naïve Bayes*.

KNN (*K-Nearest Neighbors*) adalah salah satu metode dari pengelompokan data atau klasifikasi yang menggunakan algoritma supervised learning. Algoritma ini juga merupakan algoritma partisi yang digunakan dalam pengelompokan data. KNN bekerja dengan cara menemukan k tetangga terdekat di suatu data point dan menggunakan label kelas atau nilai rata-rata tetangga tersebut sebagai prediksi untuk data point yang baru [7]. Algoritma ini bersifat non-parametrik dan dapat digunakan untuk berbagai jenis data.

Naive Bayes merupakan algoritma klasifikasi yang menggunakan dasar teorema probabilitas Bayes untuk melakukan prediksi. Algoritma ini menunjukkan hubungan antara probabilitas kondisional dari beberapa kejadian yang memanfaatkan perhitungan probabilitas untuk melakukan klasifikasi berdasarkan fitur-fitur yang dimiliki oleh suatu fitur atau data point, serta menggunakan metode probabilitas untuk membuat prediksi berdasarkan data historis yang ada. Karena

Naive Bayes sederhana, efektif, dan menghasilkan hasil yang mudah dipahami dan dapat diandalkan, alasan ini membuatnya menjadi pilihan yang populer dalam klasifikasi dan analisis prediktif [8].

Pada penelitian [3], hasil yang didapat dengan teknik klasifikasi algoritma *Naïve Bayes* untuk pembagian data menggunakan *K-Fold Cross Validation* K=10 adalah sebesar 75,78% dan untuk penggolahan menggunakan knn dengan nilai K=25 didapat hasil 74,48%. Dari hasil tersebut algoritma *naïve bayes* lebih baik dibandingkan *K-Nearest Neighbors* (KNN).

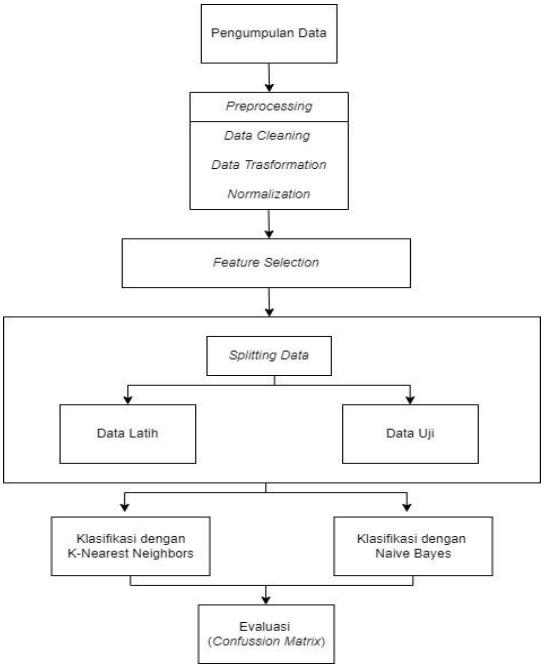
Lalu pada penelitian [9], menggunakan klasifikasi *K-Nearest Neighbor* (KNN). Hasil *sensitivity* terbaik yang didapatkan adalah 0,8125, *specificity* bernilai 0,8788 dan F1 score bernilai 0,7879 pada K = 25 dan E = 2. Hasil terbaik diperoleh dari pengujian dengan nilai fold = 4, yaitu *sensitivity* bernilai 0,6043, *specificity* bernilai 0,8703, dan F1 score bernilai 0,6383.

Dan pada penelitian [10], menggunakan *stratified K-Fold Cross Validation*. Hasil dari model terbaik terdapat pada eksperimen pertama dengan model $Y1 = 0.0438\ VAT - 0.1120$ Usia Kehamilan + 0.0887 Gula Darah + 0.0535 BMI dengan nilai *cut off* sebesar 0.0001739. Hasil pengklasifikasian dengan menggunakan k = 5 untuk semua eksperimen dan memperoleh akurasi sebesar 97.83%.

Penelitian ini bertujuan membandingkan performa dari algoritma *K-Nearest Neighbors* dan *Naïve Bayes* pada klasifikasi penyakit diabetes gestasional. Hasil penelitian ini memberikan rekomendasi kepada peneliti lain dalam menerapkan algoritma (performa) terbaik khususnya klasifikasi penyakit diabetes gestasional.

B. Metode Penelitian

Metode penelitian merupakan gambaran umum terkait tahapan proses dari penelitian yang dilakukan dari awal hingga akhir untuk mencapai tujuan penelitian. Tahapan penelitian disajikan sesuai pada Gambar 1.



Gambar 1. Metode Penelitian

1. Pengumpulan Data

Pengumpulan data merupakan tahap awal yang dilakukan pada penelitian. Data bisa didapat dari berbagai sumber, salah satu yang mudah untuk mendapatkan dataset ialah dengan mencari di internet yang dimana data tersebut merupakan data publik. Pada penelitian ini, dataset diabetes diambil dari situs Kaggle, dimana dataset ini bersifat data publik. Dataset ini berjumlah 768 data yang terdiri dari 8 fitur yaitu variabel seperti *Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age*, dan satu label yaitu *Outcome*.

2. *Preprocessing*

Preprocessing adalah tahap untuk mengolah data teks mentah menjadi data teks yang siap untuk diolah sehingga dapat masuk ketahap selanjutnya [11]. Tahap ini dapat membantu meningkatkan kualitas data yang digunakan dalam analisis dan memastikan hasil analisis yang lebih akurat [12]. Meskipun data pada penelitian ini bersifat numerik, namun untuk memastikan bahwa seluruh data telah berbentuk numerik, bersih dari *noise*, bebas dari *outliers*, dan telah distandarisasi, tahap *preprocessing* menjadi suatu langkah penting yang dilakukan.

Terdapat beberapa tahapan dalam *preprocessing* yang digunakan pada penelitian ini, yaitu sebagai berikut.

a. *Data Cleaning*
   Data Cleaning atau Pembersihan Data bertujuan untuk membersihkan data dari noise atau data yang tidak relevan seperti *missing values* (terdapat nilai yang hilang), *outliers*, data duplikat, dan data yang tidak konsisten, kemudian mempersiapkan data agar dapat diolah dan dianalisis dengan baik [13] .
b. *Data Transformation*
   Data Transformation atau Transformasi Data bertujuan untuk mengubah data asli menjadi bentuk lain sehingga data tersebut dapat memenuhi kriteria dalam sebuah penelitian [14].
c. *Normalization*
   Normalization atau Normalisasi bertujuan untuk memastikan dan mengubah nilai-nilai dalam dataset agar memiliki skala yang serupa atau terstandarisasi sehingga dapat diukur atau dibandingkan secara adil [14].

3. *Feature Selection*

Feature Selection (seleksi fitur) bertujuan untuk mengidentifikasi fitur yang paling relevan dengan kelas target [15]. *Wrapper* adalah salah satu metode *feature selection* yang umum digunakan. *Wrapper* digunakan untuk mempertimbangkan interaksi antar fitur dan mencoba berbagai kombinasi fitur untuk menemukan subset yang paling informatif. Pemilihan fitur yang baik dapat membantu mengurangi *overfitting*, meningkatkan interpretabilitas model, dan mempercepat waktu pelatihan model [16].

Salah satu teknik dari *wrapper* adalah *Forward Selection*. *Forward selection* merupakan sebuah metode *feature selection* yang dimulai dengan model kosong dan secara iteratif menambahkan satu fitur pada setiap langkahnya [17]. Setiap kali fitur ditambahkan, performa model dievaluasi dan fitur yang memberikan peningkatan performa yang signifikan dipilih untuk dimasukkan ke dalam subset fitur yang terpilih. Proses ini diulang hingga kondisi berhenti tertentu tercapai, seperti

mencapai jumlah fitur yang diinginkan atau tidak ada peningkatan performa yang signifikan lagi [16].

Dengan menerapkan *Feature Selection* menggunakan metode *Wrapper*, diharapkan penelitian ini dapat mengurangi dimensi fitur yang tidak relevan dan meningkatkan kinerja model klasifikasi dengan fokus pada fitur-fitur yang paling informatif dalam mengklasifikasi kelas target [17].

4. *Splitting* Data

Setelah melewati tahap *feature selection*, selanjutnya adalah tahap *splitting* data. *Splitting* data merupakan proses untuk membagi data menjadi dua bagian, yaitu data latih dan data uji. Tujuan dari *splitting* data adalah untuk memastikan bahwa model yang dibangun dapat digeneralisasi dengan baik pada data yang belum dilihat sebelumnya [18]. Penelitian ini dilakukan 3 kali *splitting* data yaitu 60:40, 70:30, 80:20. Data yang telah dibagi akan diuji menggunakan algoritma *K-Nearest Neighbors* dan *Naïve Bayes*.

5. Klasifikasi

Setelah dilakukan proses *splitting* data, langkah selanjutnya dalam metode penelitian ini adalah membangun model klasifikasi. Dalam penelitian ini, model klasifikasi akan dibentuk dengan menggunakan perbandingan antara *K-Nearest Neighbors* dan *Naïve Bayes*.

5.1 *K-Nearest Neighbors*

Algoritma *K-Nearest Neighbors* (K-NN atau KNN) adalah metode untuk klasifikasi objek berdasarkan data pembelajaran yang memiliki jarak paling dekat dengan objek tersebut. Keberadaan fitur-fitur yang relevan, serta bobot atau nilai penting dari setiap fitur tersebut dalam proses klasifikasi, sangat memengaruhi ketepatan algoritma K-NN [19].

Algoritma K-NN bisa dikatakan adalah algoritma pembelajaran mesin yang paling sederhana. Untuk membangun model hanya melibatkan penyimpanan dataset pelatihan. Ketika algoritma perlu membuat prediksi untuk titik data baru, langkah pertama adalah menemukan titik data terdekat dalam set data pelatihan, yang disebut sebagai "tetangga terdekatnya" [20].

Langkah-langkah cara kerja algoritma K-NN sebagai berikut :

1. Tentukan parameter K, yang merupakan bilangan bulat positif yang menunjukkan jumlah tetangga terdekat.
2. Hitung jarak antara data uji dan semua data data latih.

$$euc = \sqrt{(\sum_{i=1}^{n}(pi - qi)\,2)} \qquad \ldots\ldots (1)$$

   dimana :
   pi = sample data / data latih
   qi = data uji
   i = variabel data
   n = dimensi data
3. Tentukan K tetangga terdekat berdasarkan jarak minimum.
4. Tetangga dengan frekuensi tertinggi dianggap sebagai hasil prediksi atau klasifikasi dari data baru [21].

Langkah berikutnya adalah mengurutkan jarak dari nilai terdekat hingga terjauh. Selanjutnya, cari jumlah tetangga (nilai k) yang akan digunakan sebagai acuan untuk proses klasifikasi. Nilai k ini memungkinkan penentuan kategori dokumen berdasarkan jarak *euclidean* terdekat, di mana dokumen dianggap mirip dengan tetangga terdekatnya dan dikategorikan menurut mayoritas kategori tetangga tersebut.

5.2 *Naïve Bayes*

Naive Bayes merupakan algoritma klasifikasi yang menggunakan dasar teorema probabilitas bayes untuk melakukan prediksi. Metode ini menggunakan pendekatan probabilitas dan statistik yang dikembangkan oleh ilmuwan Inggris Thomas Bayes, yang bertujuan untuk menggunakan pengalaman masa lalu untuk memprediksi peluang di masa depan. Metode ini dikenal sebagai Teorema Bayes [19]. *Naive Bayes* menjadi pilihan yang populer dalam klasifikasi dan analisis prediktif karena sederhana, efisien, dan menghasilkan hasil yang mudah dipahami dan dapat diandalkan [8].

Berikut adalah tahapan algoritma *Naïve Bayes* :

1. Hitung jumlah kelas / label "P(H)"
2. Hitung jumlah kasus per kelas "P(X|H)"
3. Kalikan semua variabel kelas "P(X|H) * P(H)"
4. Bandingkan hasil perkelas

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)} \qquad \ldots\ldots (2)$$

Dimana :

X	: data dengan *class* yang belum diketahui
H	: hipotesis data merupakan suatu *class* spesifik
P(H|X)	: probabilitas hipotesis H berdasarkan kondisi X (*posteriori* probabilitas)
P(H)	: probabilitas hipotesis H (*prior* probabilitas)
P(X|H)	: probabilitas X berdasarkan kondisi pada hipotesis H
P(X)	: probabilitas X

Saat klasifikasi, akan memerlukan sejumlah petunjuk untuk menentukan kelas yang dianalisis. Maka akan dicari dengan menggunakan rumus :

$$Posterior = \frac{prior \,.\, likelihood}{evidence} \qquad \ldots\ldots (3)$$

dimana :

Posterior	: peluang munculnya kelas
Prior	: kelas sebelum masuknya sampel
Likelihood	: kemunculan karakteristik sampel pada kelas
Evidence	: kemunculan karakteristik sampel secara global

6. Evaluasi Model

Evaluasi model dilakukan setelah proses pengujian model selesai. Tujuan dari evaluasi dan perbandingan algoritma adalah membuktikan bahwa model benar-benar mempresentasikan sesuai pemodelan yang dilakukan dan mengukur performa dari algoritma dengan menentukan tingkat keakurasian dari dua algoritma yang digunakan [7]. Penelitian ini menggunakan hasil pengujian *confusion matrix*.

Confusion Matrix adalah tabel yang menunjukkan jumlah data uji yang benar diklasifikasikan dan yang salah diklasifikasikan. Tabel ini biasanya digunakan untuk mengevaluasi keefektifan sistem [22]. Contoh *Confusion Matrix* untuk klasifikasi biner ditunjukkan pada Tabel 2.

Tabel 1. *Confusion Matrix*

		Kelas Prediksi	
		True	*False*
Kelas Sebenarnya	*True*	TP	FN
	False	FP	TN

Rumus untuk mencari akurasi yaitu :

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \text{ X } 100\% \quad \ldots\ldots (4)$$

Keterangan :

1. *True Positive* (TP) :Model mengklasifikasikan peristiwa sebenarnya, dan peristiwa tersebut memang benar adanya.
2. *True Negative* (TN) :Model mengklasifikasikan bahwa suatu peristiwa tidak terjadi benar, dan peristiwa tersebut sebenarnya tidak terjadi.
3. *False Positive* (FP) :Model mengklasifikasikan bahwa suatu peristiwa terjadi benar, namun peristiwa tersebut sebenarnya tidak terjadi.
4. *False Negative* (FN) :Model mengklasifikasikan bahwa suatu peristiwa tidak terjadi benar, namun peristiwa tersebut sebenarnya terjadi.

C. Hasil dan Pembahasan

1. Pengumpulan Data

Pada tahap ini data yang digunakan diperoleh dari situs Kaggle yang berjumlah 768 data, berisikan 8 fitur independen dan 1 fitur dependen (label). Dataset yang digunakan disajikan pada Tabel 2.

Tabel 2. Dataset

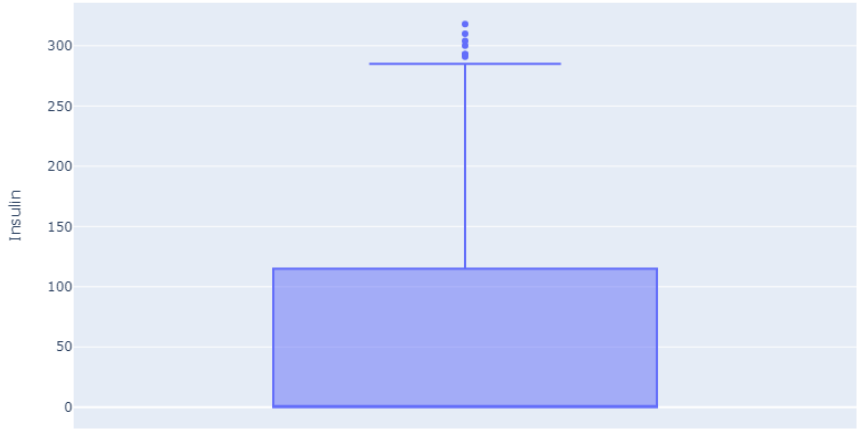
No	*Pregnancies*	*Glucose*	*Blood Pressure*	*Skin Thickness*	*Insulin*	BMI	*Diabetes Pedigree Function*	*Age*	*Outcome*
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
768	1	93	70	31	0	30.4	0.315	23	0

Pada Tabel 2 merupakan dataset diabetes yang terdiri dari fitur, yaitu *Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age*, dan *Outcome*. Berikut penjelasan masing-masing fitur :

a. *Pregnancies* adalah jumlah mengalami kehamilan.
b. *Glucose* adalah hasil gula darah.
c. *Blood Pressure* adalah hasil tekanan darah.
d. *Skin Thickness* adalah ketebalan kulit.
e. *Insulin* adalah hasil dari 2 jam suntik serum insulin.
f. BMI adalah berat dan tinggi badan pasien.
g. *Diabetes Pedigree Function* adalah nilai pengaruh riwayat penyakit diabetes.
h. *Age* adalah usia pasien.
i. *Outcome* adalah hasil mengidap diabetes atau tidak.

2. *Preprocessing*

Proses ini dilakukan tahap *data preprocessing* yaitu penanganan nilai *outliers* dengan normalisasi atau standarisasi data agar data bisa digunakan dalam proses modelling. *Ouliers* ditemukan pada salah satu fitur yaitu *Insulin*. Hasil penanganan *outliers* dapat dilihat pada Gambar 2. Gambar 2 menunjukkan distribusi dataset yang lebih terkontrol dan siap digunakan untuk proses berikutnya.



Gambar 2. Hasil Penanganan *Outliers*

3. *Feature Selection*

Data yang sudah melalui tahap *Preprocessing* akan dilanjutkan dengan proses *Feature Selection* menggunakan *wrapper forward selection*. Dalam *forward selection,* parameter K merujuk pada jumlah fitur yang ditambahkan setiap iterasi proses seleksi. K=1, berarti pada iterasi pertama, hanya satu fitur yang dipilih. K=2, berarti pada iterasi kedua, dua fitur yang dipilih. K=3, berarti pada iterasi ketiga, tiga fitur yang dipilih. K=4, berarti pada iterasi keempat, empat fitur yang dipilih. K=5, berarti pada iterasi kelima, lima fitur yang dipilih. Kemudian, *wrapper* dievaluasi menggunakan MAE, MSE dan RMSE disajikan pada Tabel 3.

Tabel 3. Evaluasi *Wrapper*

K	MAE	MSE	RMSE
K = 1	0.3493759953501142	0.1621538321754037	0.4026832901616402
K = 2	0.3370515621901651	0.1565508555039313	0.39566507996527983
K = 3	0.321702071773283	0.1449558253537637	0.3807306467225402
K = 4	0.3172771149448474	0.14234198350098234	0.3773823657434606
K = 5	0.31906365887894145	0.144061506109585070	0.3795543519834611

Berdasarkan tabel 3, iterasi yang memiliki error terendah terdapat pada K=4, dimana 4 fitur tersebut dapat dilihat pada Gambar 3.

	Pregnancies	Glucose	BMI	DiabetesPedigreeFunction
0	6	148	33.6	0.627
1	1	85	26.6	0.351
2	8	183	23.3	0.672
3	1	89	28.1	0.167
4	0	137	43.1	2.288

Gambar 3. Fitur Setelah *Wrapper*

4. *Splitting* Data

Penelitian ini melakukan 3 kali *splitting* data, yaitu 60:40, 70:30, dan 80:30. Pada *splitting* data 60:40 berarti 60% dari total data digunakan untuk data latih, dan 40% digunakan untuk data uji. Lalu, pada *splitting* data 70:30 berarti 70% dari total data digunakan untuk data latih, dan 30% digunakan untuk data uji. Dan, pada *splitting* data 80:20 berarti 80% dari total data digunakan untuk data latih, dan 20% digunakan untuk data uji. Semakin banyak atau tinggi data latih maka semakin bagus akurasinya. Data yang telah dibagi sebanyak tiga kali akan diuji menggunakan algoritma *K-Nearest Neighbor* dan *Naïve Bayes*.

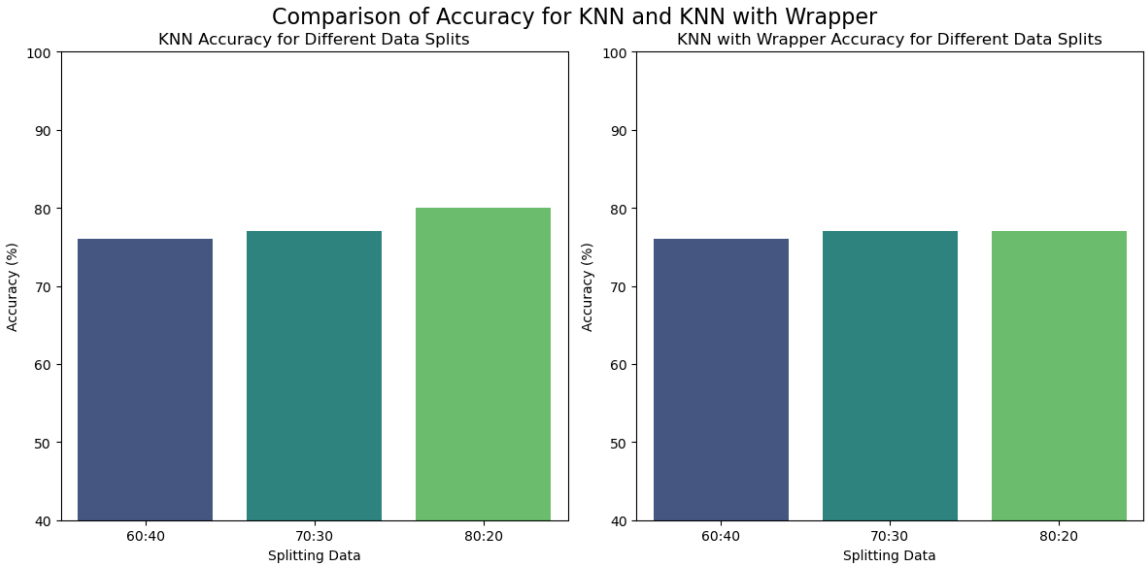
5. Klasifikasi

5.1 *K-Nearest Neighbors*

```
In [61]:
sc_x = StandardScaler()
X_train = sc_x.fit_transform(X_train)
X_test = sc_x.transform(X_test)

In [62]:
knn=KNeighborsClassifier(n_neighbors=5,metric='euclidean',p=2)
knn.fit(X_train,y_train)

KNeighborsClassifier
KNeighborsClassifier(metric='euclide
an', p=5)

In [63]:
y_pred=knn.predict(X_test)
y_pred

array([1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1,
       1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0], dtype=int64)
```

Gambar 4. *Script* Program K-NN

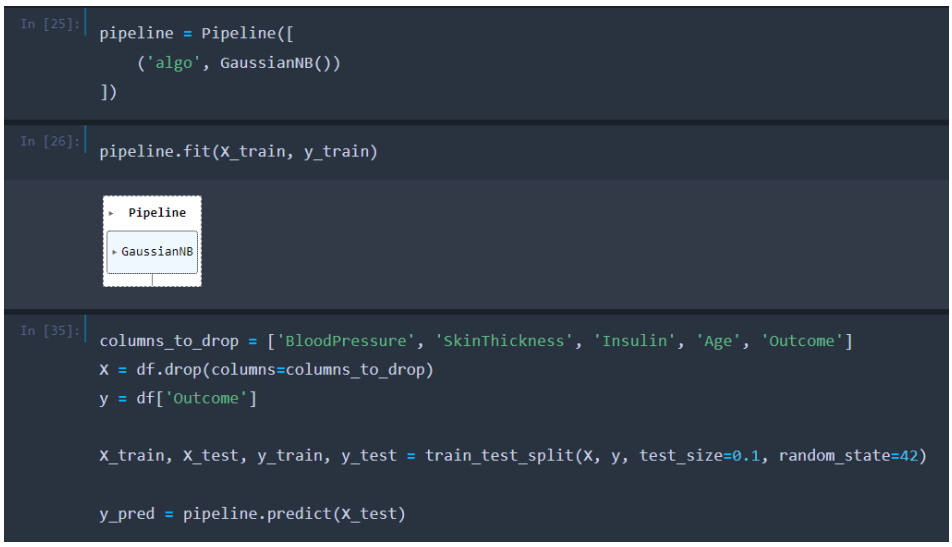
Pada Gambar 4, penggunaan *standard scaler* dari *library sckit-learn* digunakan untuk melakukan normalisasi dalam dataset. Ini merupakan tahap umum sebelum melatih model. Selanjutnya, melatih model klasifikasi menggunakan algoritma *K-Nearest Neighbors*, dan membuat prediksi pada data uji. Model *K-Nearest Neighbors* dibangun menggunakan K=5 . Hasil dari perbandingan akurasi K-NN tanpa *wrapper* dan menggunakan *wrapper* dapat dilihat pada Gambar 5.



Gambar 5. Perbandingan Akurasi K-NN Tanpa *Wrapper* dan Menggunakan W*rapper*

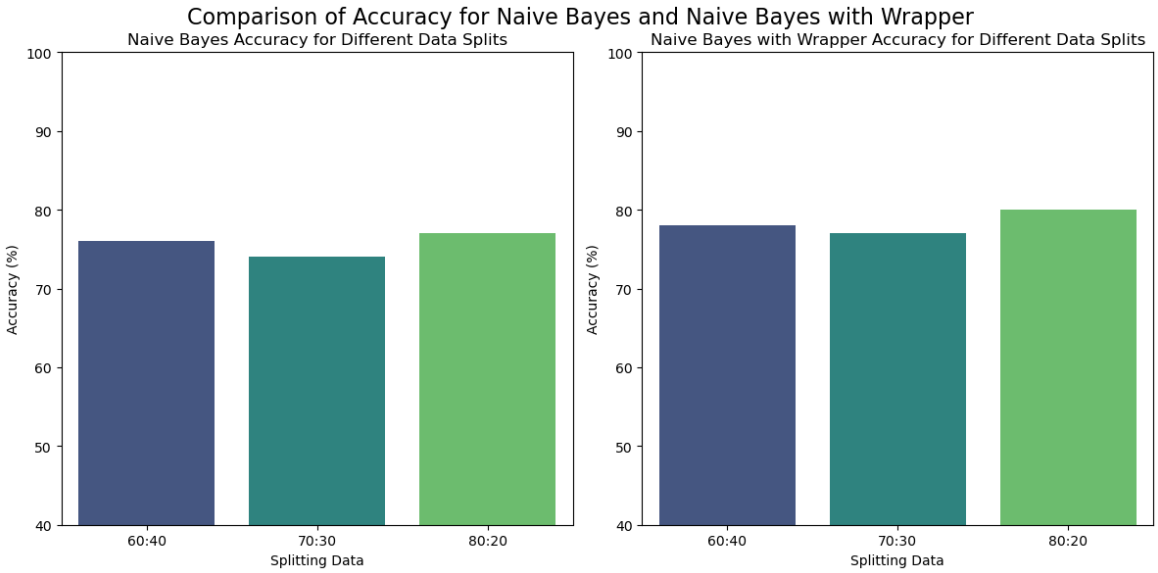
Pada gambar 5, terdapat visualisasi dari perbandingan hasil akurasi K-NN tanpa *Wrapper Feature Selection* dan menggunakan *Wrapper Feature Selection* dengan *splitting* data 60:40, 70:30, dan 80:20. Hasil akurasi K-NN dengan tiga *splitting* data, yaitu sebesar 76%, 77%, dan 80%. Sedangkan, hasil akurasi K-NN menggunakan *Wrapper Feature Selection* dengan 3 *splitting* data, yaitu sebesar 76%, 77%, dan 77%.

5.2 *Naïve Bayes*



Gambar 6. Script Program *Naïve Bayes*

Pada Gambar 6, merupakan membuat objek *Pipeline* dengan satu tahap yang disebut algo dan menggunakan model *Gaussian Naïve Bayes*. Selanjutnya, menghapus fitur tertentu dari dataset dengan menggunakan metode *drop*. Kemudian, melakukan *splitting* data menjadi data latih dan data uji. Setelah melatih *Naïve Bayes* model, selanjutnya model diuji menggunakan data uji untuk melakukan klasifikasi. Hasil dari perbandingan akurasi *Naïve Bayes* tanpa *wrapper* dan menggunakan *wrapper* dapat dilihat pada Gambar 7.



Gambar 7. Perbandingan Akurasi *Naïve Bayes* Tanpa *Wrapper* dan Menggunakan W*rapper*

Pada gambar 7, terdapat visualisasi dari perbandingan hasil akurasi *Naïve Bayes* tanpa *Wrapper Feature Selection* dan menggunakan *Wrapper Feature Selection* dengan *splitting* data 60:40, 70:30, dan 80:20. Hasil akurasi *Naïve Bayes* dengan tiga *splitting* data, yaitu sebesar 76%, 74%, dan 77%. Sedangkan, hasil akurasi *Naïve Bayes* dengan *Wrapper Feature Selection* dengan 3 *splitting* data, yaitu sebesar 78%, 77%, dan 80%.

6. Evaluasi Model

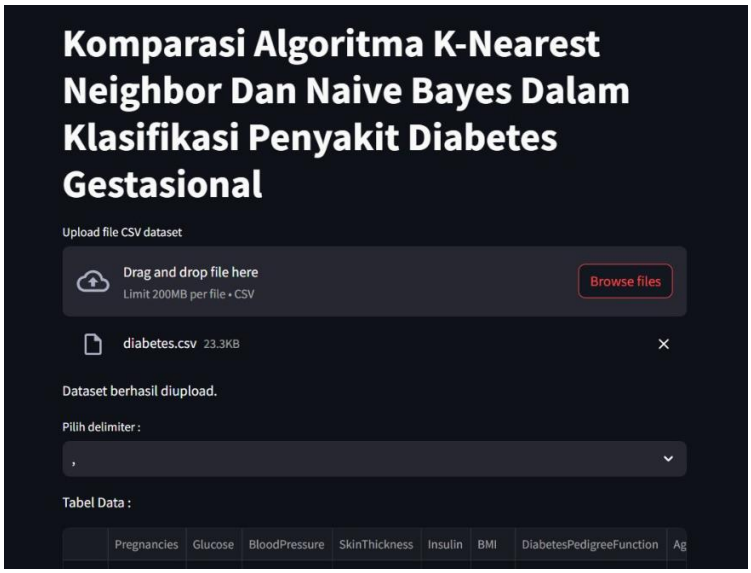
Pada penelitian menggunakan metode K-NN dan *Naïve Bayes* dengan 3 *splitting* data yaitu 60:40, 70:30, 80:20. Dengan membandingkan hasil akurasi dari K-NN dengan K = 5 dan *Naïve Bayes* tanpa *wrapper feature selection* dengan K-NN dan *Naïve Bayes* menggunakan *wrapper feature selection*. Hasil perbandingan akurasi dapat dilihat pada Tabel 4.

Tabel 4. Hasil Perbandingan

Splitting Data	Without *Wrapper Feature Selection*								With *Wrapper Feature Selection*							
	K-NN				*Naïve bayes*				K-NN				*Naïve bayes*			
	Accuracy	*Precision*	*Recall*	*F1-score*	*Accuracy*	*Precision*	*Recall*	*F1-score*	*Accuracy*	*Precision*	*Recall*	*F1-score*	*Accuracy*	*Precision*	*Recall*	*F1-score*
60 : 40	76%	73%	70%	71%	76%	73%	74%	73%	76%	73%	70%	71%	78%	75%	75%	75%
70 : 30	77%	74%	72%	73%	74%	71%	72%	72%	77%	74%	72%	73%	77%	75%	75%	75%
80 : 20	80%	76%	75%	76%	77%	75%	75%	75%	77%	72%	72%	72%	80%	78%	77%	78%

Berdasarkan Tabel 4, hasil akurasi tertinggi diperoleh pada *splitting* data 80:20 baik untuk K-NN maupun *Naïve Bayes*. Model K-NN dan *Naïve Bayes* tanpa *wrapper feature selection* masing-masing mencapai akurasi sebesar 80% dan 77%.

Sementara itu, model K-NN dan *Naïve Bayes* dengan *wrapper feature selection* masing-masing mencapai akurasi sebesar 77% dan 80%.



Gambar 8. *User Interface* dengan *Streamlit*

Pada **Gambar 8**, merupakan antarmuka pengguna (*user interface* atau UI) untuk aplikasi web. *User Interface* adalah titik interaksi antara pengguna dan sistem atau aplikasi yang dirancang untuk memudahkan navigasi, memahami informasi dan memungkinkan pengguna berinteraksi dengan aplikasi tersebut dengan cara yang mudah dan terstruktur. Sebagai salah satu contoh, dalam konteks aplikasi web adalah dengan menggunakan *Streamlit*. *Streamlit* merupakan *framework open-source* yang memungkinkan pembuatan aplikasi web interaktif dengan menggunakan *Python*. UI ini berguna untuk melihat dan memvisualisasikan data khususnya dengan menerapkan metode seperti K-NN dan *Naïve Bayes*.

D. Simpulan

Dari penelitian yang dilakukan pada dataset Diabetes Gestasional yang diambil pada Kaggle berjumlah 768 data dengan menggunakan dua algoritma yaitu K-NN dan *Naïve Bayes* diperoleh hasil performa tertinggi berada pada *splitting* data 80:20 baik K-NN maupun *Naïve Bayes*. K-NN dengan K = 5 memperoleh nilai *Accuracy* sebesar 80%, *Precision* sebesar 76%, *Recall* sebesar 75%, dan *F1-score* sebesar 76% tanpa *wrapper feature selection.* Dan, *Naïve Bayes* memperoleh nilai *Accuracy* sebesar 80%, *Precision* sebesar 78%, *Recall* sebesar 77%, dan *F1-score* sebesar 78% menggunakan *wrapper feature selection.*

Precision adalah rasio antara jumlah *true positive* (positif benar) dengan jumlah keseluruhan prediksi positif yang mengukur seberapa banyak dari prediksi positif yang sebenarnya relevan. *Recall*, atau juga disebut *sensitivity* atau *true positive rate*, adalah rasio antara jumlah *true positive* dengan jumlah keseluruhan fitur yang seharusnya positif yang mengukur seberapa banyak dari fitur yang seharusnya positif yang berhasil diidentifikasi. *F1-score* adalah *harmonic mean* dari *precision* dan *recall*. Berdasarkan hasil, dapat disimpulkan bahwa pada komparasi algoritma

KNN dan *Naïve Bayes* tanpa *wrapper feature selection,* dan KNN dan *Naïve Bayes* dengan *wrapper feature selection*, algoritma KNN tanpa *feature selection* dan *Naïve Bayes* menggunakan *feature selection* memperoleh tingkat akurasi tertinggi yang sama sebesar 80%.

E. Referensi

[1] E. Sudarmo, D. Prihanto, A. William, and J. Imbar, "Edukasi Tentang Pengendalian Diabetes Mellitus Pada Peserta Prolanis di Kota Ternate," vol. 5, no. 1, 2022.

[2] Lestari, Zulkarnain, and ST. A. Sijid, "Diabetes Melitus: Review Etiologi, Patofisiologi, Gejala, Penyebab, Cara Pemeriksaan, Cara Pengobatan dan Cara Pencegahan," 2021. [Online]. Available: http://journal.uin-alauddin.ac.id/index.php/psb

[3] B. Delvika, S. Nurhidayarnis, P. D. Rinada, N. Abror, and A. Hidayat, "Comparison of Classification Between Naive Bayes and K-Nearest Neighbor on Diabetes Risk in Pregnant Women Perbandingan Klasifikasi Antara Naive Bayes dan K-Nearest Neighbor Terhadap Resiko Diabetes Pada Ibu Hamil," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 2, no. https://journal.irpi.or.id/index.php/malcom/issue/view/17, pp. 68–75, 2022.

[4] M. Adelita, K. S. Arto, and M. Deliana, "Kontrol Metabolik pada Diabetes Melitus Tipe-1," 2020.

[5] R. Rahmaddeni and A. Kurnianto, "Data Mapping System Of Riau Province Fire Potential Using K-Means Clustering Method," 2020.

[6] A. Raup, W. Ridwan, Y. Khoeriyah, and Q. Yuliati Zaqiah, "Deep Learning dan Penerapannya dalam Pembelajaran," 2022. [Online]. Available: http://Jiip.stkipyapisdompu.ac.id

[7] K. I. F. Pratama and Kusnawi, "Komparasi Algoritma Supervised Learning dan Feature Selection pada Klasifikasi," *Indonesian Journal of Computer Science Attribution*, vol. 12, no. 6, pp. 2023–3722, 2023.

[8] E. Poerwandono and F. J. Perwitosari, "Penerapan Data Mining Untuk Penilaian Kinerja Karyawan Di PT. Riksa Dinar DJaya Menggunakan Metode Naïve Bayes Classification," *Jurnal Sains dan Teknologi*, vol. 5, no. 1, p. |pp, 2023, doi: 10.55338/saintek.v5i1.1416.

[9] V. Yolanda, I. Cholissodin, and P. P. Adikara, "Klasifikasi Diagnosis Penyakit Diabetes Gestasional pada Ibu Hamil menggunakan Algoritme Neighbor Weighted K-Nearest Neighbor (NWKNN)," 2021. [Online]. Available: http://j-ptiik.ub.ac.id

[10] S. Aisyah and A. K. N. Rifai, "Analisis Diskriminan Menggunakan Stratified K-Fold Cross Validation pada Klasifikasi Gestational Diabetes Mellitus (GDM) Ibu Hamil di Rumah Sakit XYZ," *Bandung Conference Series: Statistics*, vol. 3, no. 2, pp. 447–458, Jul. 2023, doi: 10.29313/bcss.v3i2.8366.

[11] R. Rahmaddeni and F. Akbar, "Comparison of Naïve Bayes Algorithm, Support Vector Machine and Decision Tree in Analyzing Public Opinion on COVID-19 Vaccination in Indonesia," *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 6, no. 1, p. 8, Apr. 2023, doi: 10.24014/ijaidm.v6i1.19966.

[12] L. Efrizoni, S. Defit, M. Tajuddin, and A. Anggrawan, "Komparasi Ekstraksi Fitur dalam Klasifikasi Teks Multilabel Menggunakan Algoritma Machine Learning," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 3, pp. 653–666, Jul. 2022, doi: 10.30812/matrik.v21i3.1851.

[13] F. Akbar, H. Wira Saputra, A. Karel Maulaya, and M. Fikri Hidayat, "Implementation of Decision Tree Algorithm C4.5 and Support Vector Regression for Stroke Disease Prediction Implementasi Algoritma Decision Tree C4.5 dan Support Vector Regression untuk Prediksi Penyakit Stroke," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 2, pp. 61–67, 2022.

[14] M. Z. Al-Taie, S. Kadry, and J. P. Lucas, "Online data preprocessing: A case study approach," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 2620–2626, Aug. 2019, doi: 10.11591/ijece.v9i4.pp2620-2626.

[15] R. E. Nogales and M. E. Benalcázar, "Analysis and Evaluation of Feature Selection and Feature Extraction Methods," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, Dec. 2023, doi: 10.1007/s44196-023-00319-1.

[16] F. F. Firdaus, H. A. Nugroho, and I. Soesanti, "A Review of Feature Selection and Classification Approaches for Heart Disease Prediction," 2020.

[17] Junadhi, Agustin, M. Rifqi, and M. K. Anam, "Sentiment Analysis of Online Lectures using K-Nearest Neighbors based on Feature Selection," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 3, pp. 216–225, Dec. 2022, doi: 10.23887/janapati.v11i3.51531.

[18] Q. H. Nguyen *et al.*, "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Math Probl Eng*, vol. 2021, 2021, doi: 10.1155/2021/4832864.

[19] A. M. Widodo, Y. S. Anggraeni, N. Anwar, A. Ichwani, and B. A. Sekti, "Performansi K-NN, J48, Naive Bayes dan Regresi Logistik Sebagai Algoritma Pengklasifikasi Diabetes," 2021.

[20] Raharjo, B. (2021). P (Machine Learning). Penerbit Yayasan Prima Agus Teknik, 1-110.

[21] F. T. Admojo, "Indonesian Journal of Data and Science Klasifikasi Aroma Alkohol Menggunakan Metode KNN," *Indonesian Journal of Data and Science*, vol. 1, no. https://jurnal.yoctobrain.org/index.php/ijodas/issue/view/2, pp. 34–38, 2020.

[22] A. A. Ibrahim, R. L. Ridwan, M. M. Muhammed, R. O. Abdulaziz, and G. A. Saheed, "Comparison of the CatBoost Classifier with other Machine Learning Methods," 2020. [Online]. Available: www.ijacsa.thesai.org