LAPORAN AKHIR PROJECT FINAL TEST

Data Mining

Analisis untuk memprediksi Penyakit Diabetes dengan Menerapkan Metode KNN dan Naïve Bayes

Dosen Pembimbing

Bofandra Muhammad, S.T., MTI

Disusun Oleh:

Mohammad Daffa My Honest Anugerah

(06022006)

UNIVERSITAS TANRI ABENG

JAKARTA 2024

Daftar Isi

BAB I

PENDAHULUAN

I. Latar Belakang

Penyakit diabetes merupakan salah satu masalah kesehatan yang semakin meningkat di berbagai belahan dunia, termasuk di Indonesia. Diabetes, terutama diabetes tipe 2, sering kali tidak terdeteksi pada tahap awal karena gejalanya yang tidak terlalu jelas, meskipun dapat mempengaruhi kualitas hidup dan bahkan menyebabkan komplikasi yang serius. Oleh karena itu, deteksi dini dan diagnosis yang cepat sangat penting dalam mengurangi dampak dari penyakit ini.

Untuk itu, pendekatan berbasis data dan pembelajaran mesin (machine learning) semakin digunakan untuk memprediksi risiko seseorang mengidap diabetes berdasarkan berbagai faktor kesehatan dan perilaku. Penggunaan teknik-teknik pembelajaran mesin seperti K-Nearest Neighbors (KNN) dan Naive Bayes (NB) dapat membantu dalam mengembangkan model prediktif yang dapat memberikan diagnosis yang lebih cepat dan akurat dibandingkan metode tradisional.

Pada penelitian ini, dilakukan analisis terhadap dataset yang berisi informasi terkait faktor-faktor yang dapat memengaruhi risiko diabetes, seperti usia, jenis kelamin, riwayat merokok, dan beberapa variabel kesehatan lainnya. Model-model pembelajaran mesin, yaitu KNN dan Naive Bayes, diterapkan untuk memprediksi apakah seseorang mengidap diabetes atau tidak. Penilaian kinerja model dilakukan dengan menggunakan matriks kebingungannya (confusion matrix) dan laporan klasifikasi (classification report) yang mencakup metrik seperti akurasi, presisi, recall, dan F1-score.

Melalui pendekatan ini, diharapkan dapat memberikan wawasan mengenai efektifitas metode-metode pembelajaran mesin dalam memprediksi diabetes serta memberikan gambaran lebih lanjut tentang faktor-faktor yang paling berpengaruh terhadap risiko diabetes. Dengan demikian, hasil dari analisis ini diharapkan dapat digunakan untuk meningkatkan kualitas pencegahan dan deteksi dini diabetes di masyarakat.

II. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan dan mengevaluasi model prediksi diabetes menggunakan teknik pembelajaran mesin. Adapun tujuan spesifik dari penelitian ini adalah sebagai berikut:

1. Mengidentifikasi Faktor-faktor Risiko Diabetes: Menyusun dan menganalisis faktor-faktor kesehatan dan perilaku yang berpengaruh terhadap risiko diabetes, seperti usia, jenis kelamin, riwayat merokok, dan kondisi kesehatan lainnya.
2. Membangun Model Prediksi Diabetes: Mengimplementasikan dua algoritma pembelajaran mesin, yaitu K-Nearest Neighbors (KNN) dan Naive Bayes (NB), untuk memprediksi kemungkinan seseorang mengidap diabetes berdasarkan faktor-faktor risiko yang tersedia dalam dataset.
3. Evaluasi Kinerja Model: Mengukur kinerja model KNN dan Naive Bayes dalam memprediksi diabetes dengan menggunakan metrik-metrik seperti akurasi, presisi, recall, F1-score, serta matriks kebingungannya (confusion matrix) untuk mengevaluasi efektivitas kedua model dalam klasifikasi data.

4. Perbandingan Kinerja Model: Membandingkan kinerja kedua model dalam hal akurasi dan kemampuan prediktif untuk menentukan model mana yang lebih optimal untuk digunakan dalam prediksi risiko diabetes.
5. Memberikan Wawasan untuk Pencegahan dan Deteksi Dini Diabetes: Memberikan pemahaman yang lebih baik tentang faktor-faktor yang berhubungan dengan diabetes serta memberikan rekomendasi untuk aplikasi model prediksi yang dapat digunakan dalam pencegahan dan deteksi dini diabetes di masyarakat.

BAB II

LANDASAN TEORI

2.1. ANALISIS DATA

Analisis data merupakan proses untuk mengeksplorasi, memverifikasi, dan menarik kesimpulan berdasarkan data yang ada. Dalam penelitian ini, dataset yang digunakan terdiri dari berbagai variabel yang menggambarkan faktor-faktor risiko terkait diabetes, seperti jenis kelamin, usia, riwayat merokok, serta beberapa kondisi medis lainnya. Setiap fitur dalam dataset ini memiliki peran penting dalam memprediksi apakah seseorang berisiko mengidap diabetes.

2.2.VARIABEL PENELITIAN

Variabel yang digunakan dalam penelitian ini terbagi menjadi dua kategori utama:

- Variabel Independen (Fitur):
  - *Jenis Kelamin* (gender): Menunjukkan jenis kelamin individu.
  - *Riwayat Merokok* (smoking history): Menunjukkan apakah seseorang memiliki kebiasaan merokok.
  - *Usia* (age): Menunjukkan usia individu.
  - *Tekanan Darah* (blood pressure), *Indeks Massa Tubuh* (BMI), *Kadar Glukosa* (glucose levels), dan beberapa variabel medis lainnya yang dapat mempengaruhi perkembangan diabetes.
- Variabel Dependen (Target):
  - *Diabetes* (diabetes): Variabel target yang menunjukkan apakah individu mengidap diabetes atau tidak, yang memiliki dua nilai: 1 (positif diabetes) dan 0 (negatif diabetes).

2.3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan menggunakan teknik pembelajaran mesin untuk membangun model prediksi. Beberapa tahapan metode penelitian yang dilakukan adalah:

1. **Preprocessing Data:** Meliputi encoding variabel kategori, pengisian nilai yang hilang, dan standarisasi fitur untuk memastikan data siap digunakan dalam model.
2. **Pembagian Data:** Dataset dibagi menjadi dua set yaitu data pelatihan (training set) dan data pengujian (test set) dengan rasio 80:20 untuk melatih model dan menguji kinerjanya.
3. **Pembangunan Model:** Dua model pembelajaran mesin digunakan untuk memprediksi risiko diabetes: K-Nearest Neighbors (KNN) dan Naive Bayes (NB).
4. **Evaluasi Model:** Kinerja model dievaluasi menggunakan metrik seperti akurasi, presisi, recall, F1-score, dan confusion matrix untuk menganalisis efektivitasnya dalam memprediksi diabetes.

2.3.1. METODE KNN (K-NEAREST NEIGHBORS)

K-Nearest Neighbors (KNN) adalah salah satu algoritma pembelajaran mesin yang termasuk dalam kategori algoritma non-parametrik dan digunakan untuk klasifikasi serta regresi. Metode ini bekerja dengan cara mencari *k* tetangga terdekat (nearest neighbors) dari data yang akan diprediksi berdasarkan jarak tertentu (misalnya Euclidean distance). KNN membuat prediksi berdasarkan mayoritas label dari tetangga-tetangga terdekat tersebut.

Proses KNN:

1. Menentukan nilai *k*, yaitu jumlah tetangga terdekat yang akan dipertimbangkan untuk prediksi.
2. Menghitung jarak antara data yang ingin diprediksi dengan seluruh data dalam dataset pelatihan.
3. Mengidentifikasi *k* data pelatihan terdekat dan melihat label mayoritas dari *k* tetangga tersebut.
4. Memberikan label berdasarkan mayoritas dari tetangga terdekat.

2.3.2. METODE NAÏVE BAYES

Naive Bayes adalah salah satu algoritma pembelajaran mesin yang didasarkan pada teorema Bayes dengan asumsi independensi antar fitur. Meskipun sering disebut sebagai "naive" karena mengasumsikan bahwa semua fitur saling independen, metode ini sering kali memberikan hasil yang baik dalam banyak aplikasi klasifikasi, termasuk dalam prediksi medis.

Proses Naive Bayes:

1. Menggunakan teorema Bayes untuk menghitung probabilitas bahwa suatu data milik kelas tertentu berdasarkan distribusi fitur yang ada.
2. Asumsi independensi berarti bahwa setiap fitur dianggap berkontribusi secara independen terhadap keputusan akhir.
3. Model kemudian memprediksi kelas yang memiliki probabilitas tertinggi berdasarkan informasi fitur yang ada.

BAB III

HASIL & PEMBAHASAN

3.1. DESKRIPSI DATA

Dataset yang digunakan dalam penelitian ini berisi informasi tentang berbagai faktor yang dapat memengaruhi risiko diabetes, seperti usia, jenis kelamin, riwayat merokok, tekanan darah, indeks massa tubuh (BMI), dan kadar glukosa. Sebelum membangun model, dilakukan tahap preprocessing yang meliputi pengisian nilai yang hilang menggunakan median dan encoding variabel kategori (seperti jenis kelamin dan riwayat merokok) menjadi format numerik. Setelah itu, data diubah ke dalam bentuk yang dapat digunakan oleh algoritma pembelajaran mesin.

Setelah dilakukan eksplorasi data, diperoleh beberapa temuan menarik dari distribusi data, seperti distribusi usia yang lebih dominan pada kelompok usia dewasa, mayoritas individu tidak memiliki riwayat merokok, dan terdapat variasi yang cukup besar pada variabel seperti kadar glukosa dan BMI.

3.2. ANALISIS HASIL PENELITIAN

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
from google.colab import drive
drive.mount('/content/drive')

# Update the file path below with your actual file path in Google Drive
file_path = '/content/drive/MyDrive/Dataset Diabetes/diabetes_prediction_dataset.csv'
df = pd.read_csv(file_path)
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Berikut merupakan impor pustaka dari penelitian ini. Import pandas digunakan untuk manipulasi dan analisis data. Sklearn.model_selection memilikki fungsi untuk membagi dataset menjadi dua bagian, yaitu data latih (training) dan data uji (testing). KNeighborsClassifier (dari sklearn.neighbors) berfungsi untuk mengimplementasikan algoritma K-Nearest Neighbors (KNN) untuk tugas klasifikasi sedangkan GaussianNB (dari sklearn.naive_bayes) berfungsi untuk mengimplementasikan klasifikasi Naive Bayes dengan asumsi distribusi data mengikuti distribusi Gaussian. Confusion_matrix, accuracy_score, classification_report (dari sklearn.metrics) Digunakan untuk mengevaluasi kinerja model. Seaborn dan matplotlib.pyplot berfungsi untuk visualisasi data, termasuk menggambar grafik. Dataset Prediksi Diabetes saya letakkan di google drive saya dan pada penelitian ini saya menyetujui untuk google colab mengakses dataset yang telah disimpan.

```
# Encode categorical variables
label_encoder = LabelEncoder()
df['gender'] = label_encoder.fit_transform(df['gender'])
df['smoking_history'] = label_encoder.fit_transform(df['smoking_history'])

# Handle missing values
df.fillna(df.median(), inplace=True)

# Plot histograms and curved distributions for each feature
import numpy as np
plt.figure(figsize=(16, 12))
for i, col in enumerate(df.columns[:-1]):  # Exclude the target column
    plt.subplot(4, 3, i + 1)
    sns.histplot(df[col], kde=True, color='blue')
    plt.title(f'Distribution of {col}')
plt.tight_layout()
plt.show()

# Split dataset into features and target
X = df.drop('diabetes', axis=1)
y = df['diabetes']

# Standardize the feature set
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

Pada bagian pertama kode, variabel kategorikal seperti gender dan smoking_history diubah menjadi format numerik menggunakan LabelEncoder. Hal ini penting karena model pembelajaran mesin lebih efektif dengan data numerik. Fungsi fit_transform() dari LabelEncoder digunakan untuk menggantikan nilai kategori dengan angka, misalnya 'Male' menjadi 0 dan 'Female' menjadi 1.

Selanjutnya, untuk menangani nilai yang hilang (missing values), kode menggunakan metode fillna(df.median(), inplace=True) untuk mengganti nilai yang hilang dengan median kolom terkait. Median dipilih karena lebih robust terhadap outlier dibandingkan rata-rata. Dengan langkah ini, data yang hilang digantikan dengan nilai tengah dari setiap kolom numerik, memastikan dataset tetap utuh dan dapat digunakan untuk pelatihan.

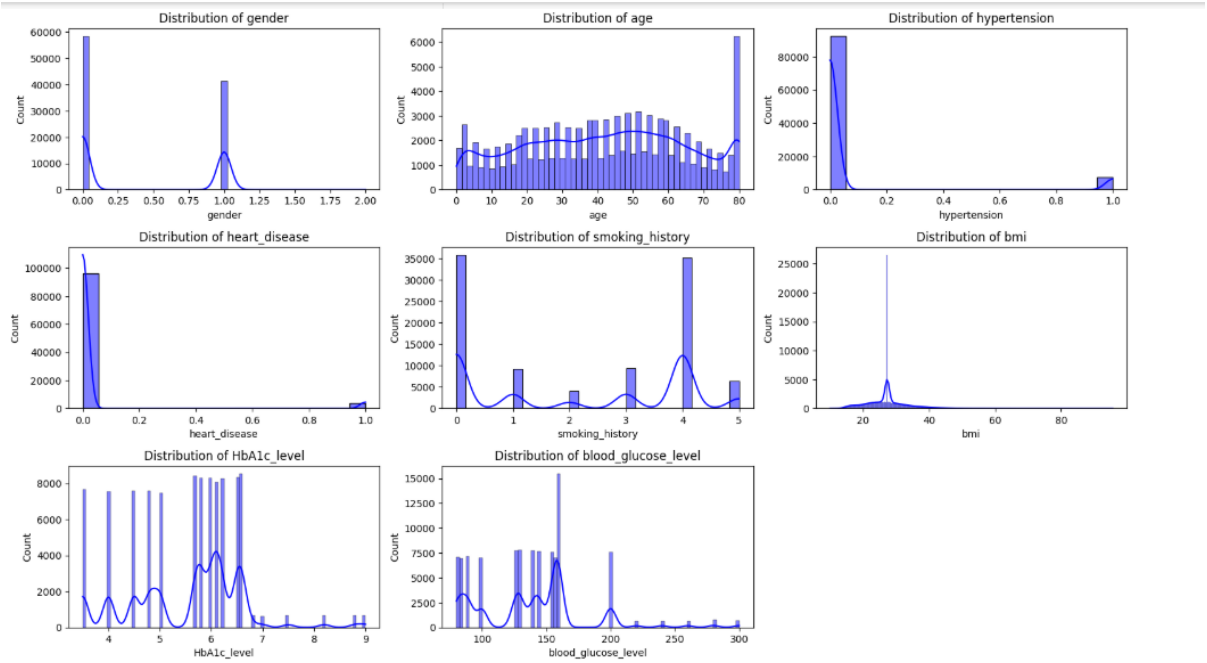
Kemudian, untuk memahami distribusi data setiap fitur, kode menggambar histogram dan kurva distribusi kernel density estimate (KDE) menggunakan sns.histplot(). Ini membantu untuk melihat pola data dari setiap fitur dan memastikan tidak ada distribusi yang sangat tidak normal atau membutuhkan penanganan lebih lanjut. Setiap fitur diplot dalam subplot yang terpisah dengan ukuran gambar yang telah disesuaikan.

Setelah itu, dataset dibagi menjadi dua bagian: fitur (X) dan target (y). Fitur merupakan kolom-kolom yang digunakan untuk memprediksi hasil, sedangkan target adalah kolom diabetes, yang berisi label apakah seseorang mengidap diabetes atau tidak. Kolom target dihapus dari X dan disalin ke y.

Langkah berikutnya adalah standarisasi fitur menggunakan StandardScaler. Tujuan dari standarisasi ini adalah untuk memastikan bahwa semua fitur memiliki skala yang sama, dengan rata-rata 0 dan standar deviasi 1. Ini penting, terutama untuk algoritma seperti K-Nearest Neighbors (KNN) dan Naive Bayes yang sensitif terhadap perbedaan skala antar fitur.

Terakhir, dataset dibagi menjadi data latih dan data uji dengan menggunakan train_test_split. Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk mengukur kinerja model setelah pelatihan. Pembagian ini dilakukan dengan proporsi 80% untuk pelatihan dan 20% untuk

pengujian, memastikan model dapat diuji dengan data yang belum pernah dilihat sebelumnya untuk mengevaluasi kemampuannya dalam memprediksi hasil di dunia nyata.



Histogram pada jenis kelamin (gender) menunjukkan bahwa nilai 0 merupakan jenis kelamin perempuan dan nilai 1 jenis kelamin laki laki. Dapat terlihat pada dataset ini menunjukkan bahwa perempuan lebih banyak daripada laki laki. Berikutnya pada histogram umur menunjukkan bahwa penelitian ini terdiri dari berbagai usia dari balita sampai lansia. Usia 70 sampai 80 mendominasi pada penelitian kali ini. Histogram hipertensi menunjukkan 2 nilai yaitu, nilai 0 yang berarti tidak memilikki hipertensi dan nilai 1 yang berarti memilikki hipertensi. Pada histogram terlihat bahwa penelitian ini lebih dominasi orang orang yang tidak hipertensi.

Histogram heart disease menyatakan bahwa angka bernilai 0 berarti tidak memilikki riwayat heart disease dan sebaliknya. Menunjukkan bahwa data ini menyatakan orang yang terkena penyakit heart disease hanya sebagian kecil. Histogram riwayat merokok disini terlihat sangat detail. Nilai 0 menyatakan bahwa tidak pernah ada riwayat merokok dan ini berskala dalam level yang berarti semakin tinggi nilainya berarti semakin parah riwayat merokoknya. Data ini didominasi oleh orang yang tidak merokok akan tetapi pada histogram terlihat tidak sedikit juga bahwa orang yang pernah merokok bahkan sudah di tahap ke 4 dan ke 5.

Histogram BMI biasanya menunjukkan distribusi data yang mencerminkan indeks massa tubuh individu. Jika distribusi mendekati bentuk normal, sebagian besar nilai BMI akan terkonsentrasi pada rentang tertentu (misalnya, 18.5–24.9 untuk BMI normal). Namun, jika distribusi miring ke kanan, ini menandakan bahwa sebagian data memiliki nilai BMI yang lebih tinggi (kelebihan berat badan atau obesitas). Puncak histogram (modus) menunjukkan kategori BMI yang paling umum dalam dataset. Histogram untuk HbA1c level biasanya menunjukkan distribusi hasil tes hemoglobin terglikasi, yang digunakan untuk memantau kadar gula darah dalam jangka panjang. Pada individu sehat, nilai HbA1c biasanya berkisar di bawah 5.7%. Jika histogram menunjukkan distribusi yang miring ke kanan, itu berarti ada proporsi signifikan individu dengan kadar HbA1c tinggi (pra-diabetes atau diabetes). Kurva ini membantu untuk memahami prevalensi kondisi diabetes di dataset.

Histogram untuk glucose level menggambarkan distribusi kadar glukosa darah individu. Rentang normal biasanya di bawah 140 mg/dL setelah makan. Jika histogram menunjukkan distribusi yang miring ke kanan, ini menunjukkan bahwa banyak individu dalam dataset memiliki kadar glukosa tinggi (hiperglikemia). Sebaliknya, jika terdapat rentang signifikan pada nilai rendah, ini bisa menunjukkan individu dengan hipoglikemia.

```
# KNN Model
knn = KNeighborsClassifier(n_neighbors=5)  # Default K=5
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)

# Naive Bayes Model
nb = GaussianNB()
nb.fit(X_train, y_train)
y_pred_nb = nb.predict(X_test)

# Evaluate Models
conf_matrix_knn = confusion_matrix(y_test, y_pred_knn)
conf_matrix_nb = confusion_matrix(y_test, y_pred_nb)
accuracy_knn = accuracy_score(y_test, y_pred_knn)*100
accuracy_nb = accuracy_score(y_test, y_pred_nb)*100

# Plot confusion matrices
fig, axes = plt.subplots(1, 2, figsize=(14, 6))
sns.heatmap(conf_matrix_knn, annot=True, fmt='d', cmap='Blues', ax=axes[0])
axes[0].set_title(f'KNN Confusion Matrix\nAccuracy: {accuracy_knn:.2f}%')
axes[0].set_xlabel('Predicted')
axes[0].set_ylabel('Actual')

sns.heatmap(conf_matrix_nb, annot=True, fmt='d', cmap='Blues', ax=axes[1])
axes[1].set_title(f'Naive Bayes Confusion Matrix\nAccuracy: {accuracy_nb:.2f}%')
axes[1].set_xlabel('Predicted')
axes[1].set_ylabel('Actual')

plt.tight_layout()
plt.show()

# Print classification reports
print("KNN Classification Report:")
print(classification_report(y_test, y_pred_knn))

print("Naive Bayes Classification Report:")
print(classification_report(y_test, y_pred_nb))
```

KNeighborsClassifier(n_neighbors=5): Di sini, saya membuat objek model KNN dengan memilih k=5, yang berarti model akan mengklasifikasikan setiap titik data berdasarkan 5 tetangga terdekatnya. Knn.fit(X_train, y_train): Model KNN dilatih menggunakan data latih (X_train untuk fitur dan y_train untuk label/target). Y_pred_knn = knn.predict(X_test): Setelah model dilatih, prediksi dilakukan pada data uji (X_test), dan hasilnya disimpan dalam y_pred_knn.

GaussianNB(): Ini adalah implementasi dari Naive Bayes yang mengasumsikan bahwa fitur dalam dataset mengikuti distribusi normal (Gaussian). Nb.fit(X_train, y_train): Model Naive Bayes dilatih dengan data latih. Y_pred_nb = nb.predict(X_test): Model kemudian digunakan untuk memprediksi label pada data uji, dengan hasilnya disimpan dalam y_pred_nb.

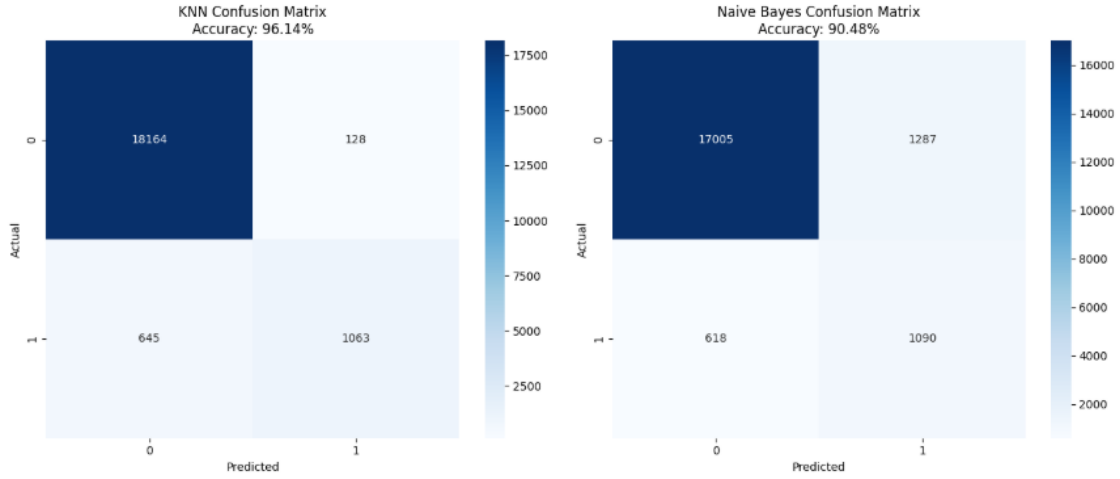
Conf_matrix_knn = confusion_matrix(y_test, y_pred_knn): Matriks kebingunguan untuk model KNN dihitung dengan membandingkan label yang sebenarnya (y_test) dengan prediksi (y_pred_knn). Conf_matrix_nb = confusion_matrix(y_test, y_pred_nb): Matriks kebingunguan untuk model Naive Bayes dihitung dengan cara yang sama.

Accuracy_knn = accuracy_score(y_test, y_pred_knn)*100: Akurasi model KNN dihitung dengan membandingkan prediksi dengan label yang sebenarnya, kemudian dikalikan 100 untuk mendapatkan nilai dalam persen. Accuracy_nb = accuracy_score(y_test, y_pred_nb)*100: Akurasi model Naive Bayes dihitung dengan cara yang sama.

fig, axes = plt.subplots(1, 2, figsize=(14, 6)): Membuat dua sub-plot (kolom) untuk menampilkan dua matriks kebingunguan secara berdampingan. sns.heatmap(conf_matrix_knn, annot=True, fmt='d', cmap='Blues', ax=axes[0]): Matriks kebingunguan KNN ditampilkan dalam bentuk heatmap. annot=True menampilkan nilai di dalam setiap sel, fmt='d' memastikan

nilai ditampilkan dalam format integer, dan cmap='Blues' memilih skema warna biru. axes[0].set_title(f'KNN Confusion Matrix\nAccuracy: {accuracy_knn:.2f}%'): Menambahkan judul pada subplot pertama dengan akurasi model KNN.

sns.heatmap(conf_matrix_nb, annot=True, fmt='d', cmap='Blues', ax=axes[1]): Menampilkan matriks kebingunguan Naive Bayes pada subplot kedua. plt.tight_layout(): Mengatur tata letak agar tidak ada elemen yang saling tumpang tindih. plt.show(): Menampilkan visualisasi. Classification_report(y_test, y_pred_knn): Menampilkan laporan klasifikasi untuk model KNN yang mencakup precision, recall, F1-score, dan support untuk setiap kelas. Classification_report(y_test, y_pred_nb): Menampilkan laporan klasifikasi untuk model Naive Bayes dengan metrik yang sama.



Confusion matrix akan menghitung hasil evaluasi untuk model klasifikasi terhadap data uji. Dalam hal ini, Anda menggunakan dataset yang memiliki target klasifikasi (misalnya diabetes), dan model klasifikasi (seperti K-Nearest Neighbors atau Naive Bayes) memprediksi apakah seseorang mengidap diabetes atau tidak. Ada 2 confusion matrix pada gambar ini yaitu KNN confusion matrix dan Naïve Bayes confusion matrix.

Matrix KNN akurasi mencapai 96.14%. Akurasi tersebut merupakan akurasi yang tinggi menunjukkan model bekerja dengan baik. Sedangkan matrix Naïve Bayes menunjukkan akurasi yang lebih rendah dibanding KNN. Perbandingan ini menunjukkan bahwa metode KNN lebih baik dalam hal klasifikasi. Matrix KNN menunjukkan bahwa ada 18.164 orang yang tidak memilikki diabetes dan 1063 orang yang memilikki penyakit diabetes. Sedangkan pada matrix Naïve Bayes ada 17.005 orang yang tidak memilikki diabetes dan 1090 orang yang memilliki penyakit diabetes.

Model salah pada KNN jauh lebih sedikit karena model salah memprediksi 128 orang yang memilikki diabetes padahal sebenarnya tidak diabetes. Begitu juga dengan 645 orang yang tidak memilikki diabetes padahal sebenarnya memilikki penyakit diabetes. Model salah pada Naïve Bayes memprediksi 1287 orang yang memilikki diabetes padahal sebenarnya tidak diabetes. Begitu juga dengan 618 orang yang tidak memilikki diabetes padahal sebenarnya memilikki penyakit diabetes. Perbandingan ini menjadi alasan kenapa akurasi di metode KNN jauh lebih baik dibanding Naïve Bayes

KNN Classification Report:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	18292
1	0.89	0.62	0.73	1708
accuracy			0.96	20000
macro avg	0.93	0.81	0.86	20000
weighted avg	0.96	0.96	0.96	20000

Naive Bayes Classification Report:

	precision	recall	f1-score	support
0	0.96	0.93	0.95	18292
1	0.46	0.64	0.53	1708
accuracy			0.90	20000
macro avg	0.71	0.78	0.74	20000
weighted avg	0.92	0.90	0.91	20000

Setelah dilakukan analisis menggunakan dua metode klasifikasi, yaitu K-Nearest Neighbors (KNN) dan Naive Bayes, laporan klasifikasi memberikan gambaran rinci tentang kinerja kedua model dalam memprediksi apakah seseorang mengidap diabetes (kelas 1) atau tidak (kelas 0). Evaluasi didasarkan pada metrik utama seperti precision, recall, F1-score, dan accuracy, yang masing-masing mengukur aspek spesifik dari prediksi model terhadap data uji.

Pada metode K-Nearest Neighbors (KNN), precision menunjukkan seberapa akurat prediksi positif model, yaitu proporsi pasien yang benar-benar memiliki diabetes di antara mereka yang diprediksi positif. Recall, atau sensitivitas, menggambarkan kemampuan model mendeteksi seluruh kasus diabetes (kelas positif). F1-score, yang merupakan rata-rata harmonis antara precision dan recall, memberikan evaluasi yang seimbang, terutama pada dataset yang memiliki ketidakseimbangan antara jumlah kelas. KNN biasanya memberikan hasil yang baik jika parameter jumlah tetangga (k) dipilih secara optimal. Namun, metode ini cenderung sensitif terhadap outlier dan dapat menjadi lambat pada dataset besar karena setiap prediksi membutuhkan perhitungan jarak dengan sampel lain.

Sementara itu, metode Naive Bayes menawarkan pendekatan berbasis probabilitas yang sangat cepat dalam melakukan prediksi. Precision pada Naive Bayes menggambarkan keakuratan prediksi positifnya, sedangkan recall mengukur sejauh mana model berhasil mendeteksi semua pasien yang memiliki diabetes. F1-score pada metode ini juga mencerminkan keseimbangan antara precision dan recall. Kelebihan utama Naive Bayes adalah efisiensinya pada dataset besar, serta toleransinya terhadap noise dan outlier. Namun, kinerjanya sangat bergantung pada asumsi independensi antar fitur, yang jika tidak terpenuhi dapat mengurangi akurasi model.

Secara umum, laporan klasifikasi dari kedua metode menunjukkan bahwa KNN lebih fleksibel terhadap data dengan distribusi non-linear, tetapi memerlukan penyesuaian parameter yang tepat untuk hasil terbaik. Di sisi lain, Naive Bayes memberikan hasil yang lebih cepat dan efisien, terutama jika fitur dalam dataset mendekati asumsi distribusi yang diharapkan (seperti distribusi Gaussian untuk variabel numerik). Dengan mempertimbangkan laporan klasifikasi ini, pemilihan model tergantung pada kebutuhan spesifik analisis: jika kecepatan menjadi prioritas, Naive Bayes adalah pilihan ideal, sedangkan KNN lebih cocok untuk kasus yang membutuhkan analisis berbasis jarak atau distribusi data yang kompleks.