Laporan Mata Kuliah Pembelajaran Mesin

TUGAS 3

disusun untuk memenuhi tugas mata kuliah Pembelajaran Mesin

Oleh :

Kelompok 3

Anggota :

Meutia Aini	(2208107010005)
Akhsania Maisa Rahmah	(2208107010017)
Fadli Ahmad Yazid	(2208107010032)
Muhammad Mahathir	(2208107010056)
Muhammad Aufa Zaikra	(2208107010070)

JURUSAN INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SYIAH KUALA

DARUSSALAM, BANDA ACEH

2025

A. Data Description

Dataset **Diabetes Prediction** yang tersedia di Kaggle dan dibuat oleh pengguna *iammustafatz* merupakan kumpulan data kesehatan yang dirancang untuk memprediksi kemungkinan seseorang menderita diabetes berdasarkan beberapa indikator klinis dan gaya hidup. Dataset ini berisi **100.000 entri data pasien** dengan **9 atribut**, termasuk usia, jenis kelamin, riwayat hipertensi dan penyakit jantung, riwayat merokok, indeks massa tubuh (BMI), kadar hemoglobin A1c, dan kadar glukosa darah. Target prediksi berupa kolom `diabetes` yang bernilai 1 jika pasien menderita diabetes dan 0 jika tidak. Dataset ini bersih (tanpa missing value), dan disusun untuk mendukung eksperimen dalam klasifikasi biner menggunakan algoritma supervised learning. Dengan kombinasi variabel yang cukup beragam, dataset ini cocok digunakan untuk eksplorasi machine learning di bidang kesehatan, terutama dalam deteksi dini penyakit diabetes.

Pemilihan dataset Diabetes Prediction sudah memenuhi ketentuan tugas sebagai berikut: Dataset bersumber dari Kaggle, platform terpercaya untuk dataset machine learning, tersedia di Diabetes Prediction Dataset. Dataset ini terakhir diperbarui pada tahun 2023. Dataset telah digunakan dalam penelitian yang dipublikasikan, termasuk oleh Kaliappan et al. (2024), Alzubaidi et al. (2024), dan Alzubaidi et al. (2023), yang mengkonfirmasi relevansinya untuk prediksi diabetes. Dataset memiliki 9 atribut (usia, jenis kelamin, hipertensi, penyakit jantung, riwayat merokok, BMI, kadar HbA1c, kadar glukosa darah, dan status diabetes) dengan target biner (0 untuk non-diabetes, 1 untuk diabetes), yang dijelaskan secara rinci pada bagian "Data Understanding". Dengan demikian, dataset ini mendukung eksperimen klasifikasi menggunakan algoritma Logistic Regression, K-Nearest Neighbor, Naïve Bayes, dan Decision Tree.

B. Data Loading

Pada bagian ini memuat dataset Diabetes Prediction ke dalam lingkungan pemrograman Python. Proses ini menggunakan library Pandas untuk memuat dan memanipulasi data. Dataset menggunakan fungsi read_csv() dari Pandas. Dataset tersedia dalam format CSV, sehingga mudah dimuat menggunakan Pandas. Contoh kode nya yaitu:

```
# Membaca dataset dari sumber
df = pd.read_csv('diabetes_prediction_dataset.csv')

# Menampilkan 5 data pertama
print("Data Awal:")
display(df.head())
```

C. Data Understanding

1. Informasi Dataset

```
Informasi Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   gender               100000 non-null  object
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64
 3   heart_disease        100000 non-null  int64
 4   smoking_history      100000 non-null  object
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64
 8   diabetes             100000 non-null  int64
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
```

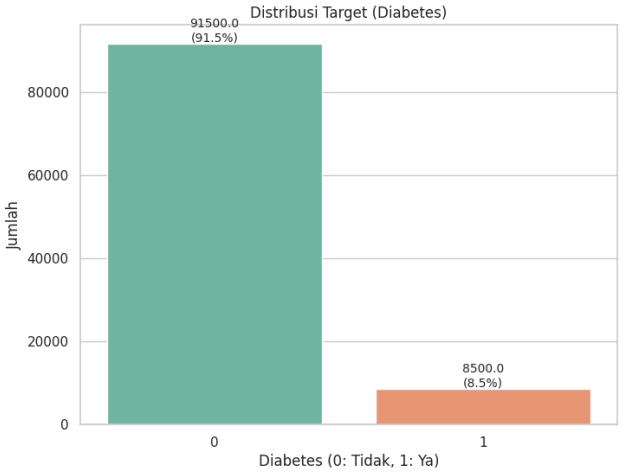
Dapat kita lihat bahwa dataset memiliki 9 kolom dengan 3 kolom bertipe float64, 4 kolom bertipe int64, dan 2 kolom bertipe object. Semua kolom memiliki 100.000 data non-null, artinya tidak ada nilai yang hilang.

```
# Memeriksa jumlah nilai duplikat
duplicates = df.duplicated().sum()
print(f"Jumlah Data Duplikat: {duplicates}")
```

```
Jumlah Data Duplikat: 3854
```

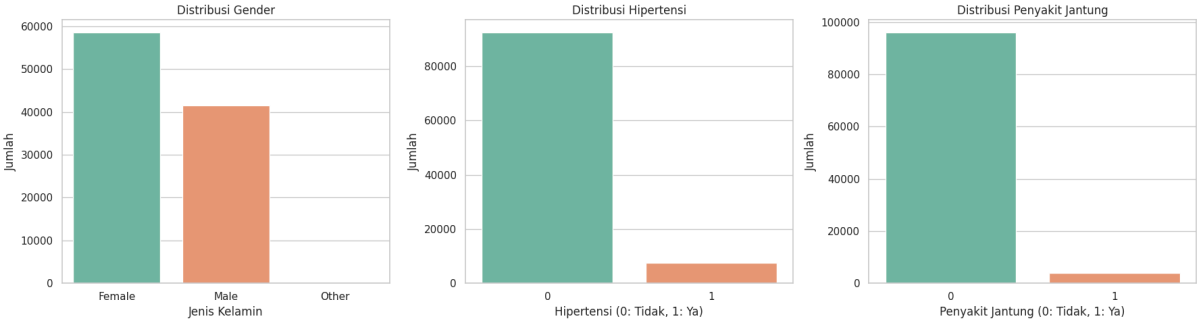
Dataset memiliki 3.854 data duplikat yang perlu ditangani.

2. Visualisasi Data
   - Distribusi Target (Diabetes)



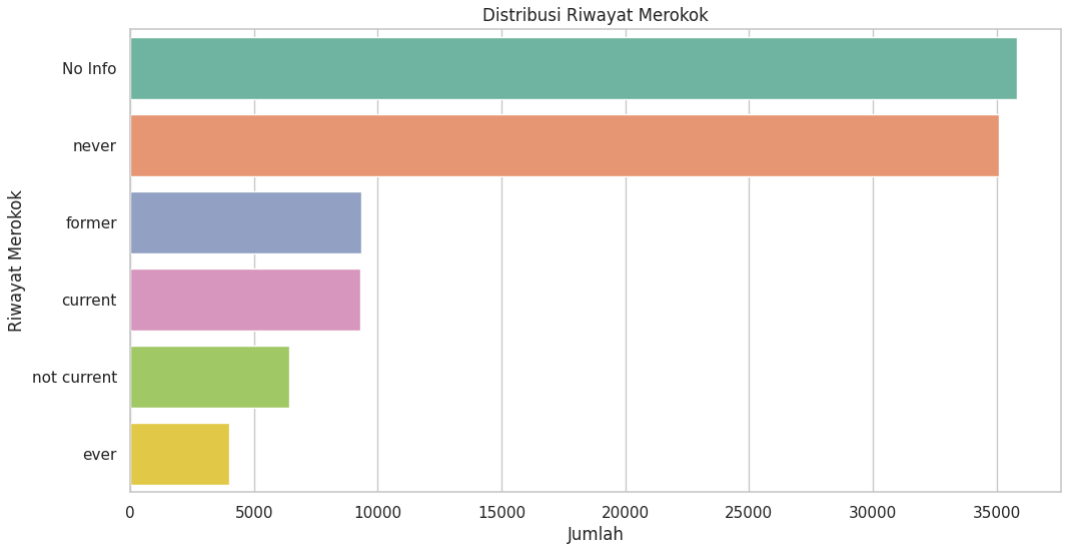
Visualisasi distribusi target menunjukkan bahwa dataset tidak seimbang, dengan jumlah data untuk kelas tidak diabetes jauh lebih banyak dibandingkan kelas diabetes.

- Distribusi Fitur Kategorikal



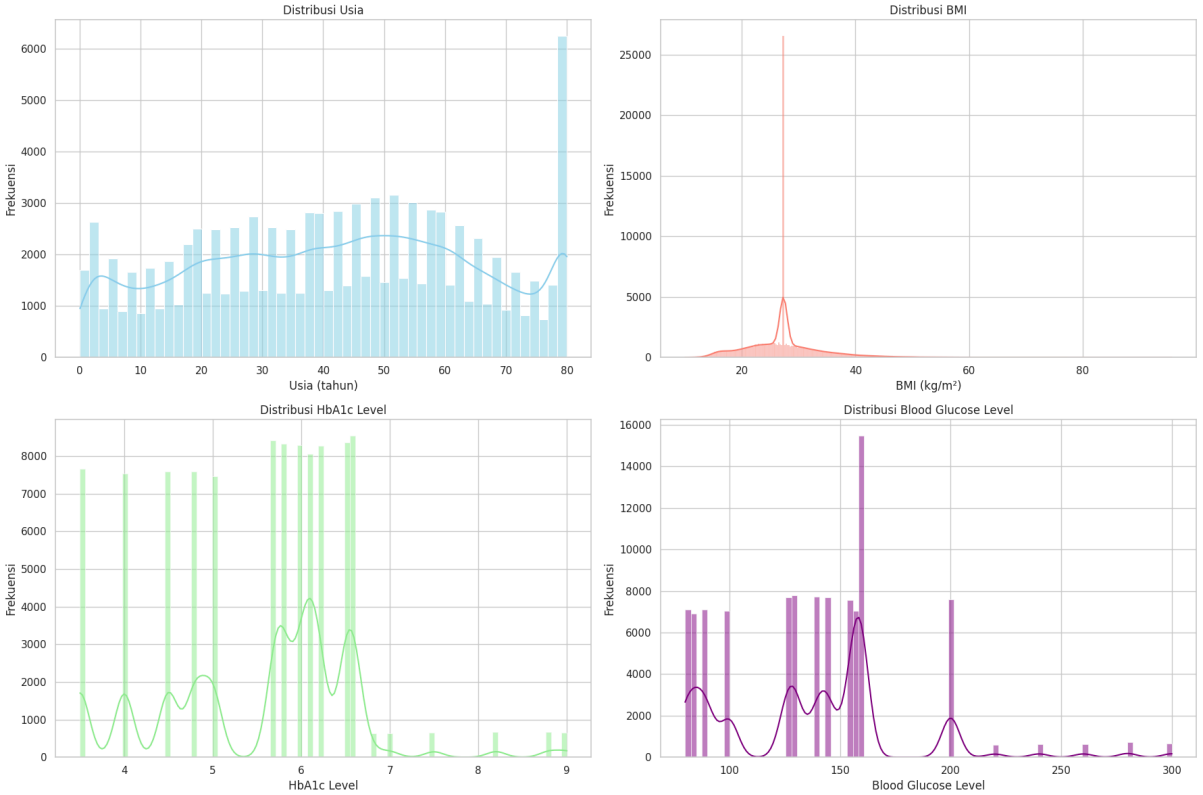
Dari visualisasi fitur kategorikal, kita dapat melihat distribusi gender, hipertensi, dan penyakit jantung dalam dataset. Distribusi gender menunjukkan mayoritas data adalah Female, sementara mayoritas data tidak memiliki hipertensi atau penyakit jantung.

- Distribusi Riwayat Merokok



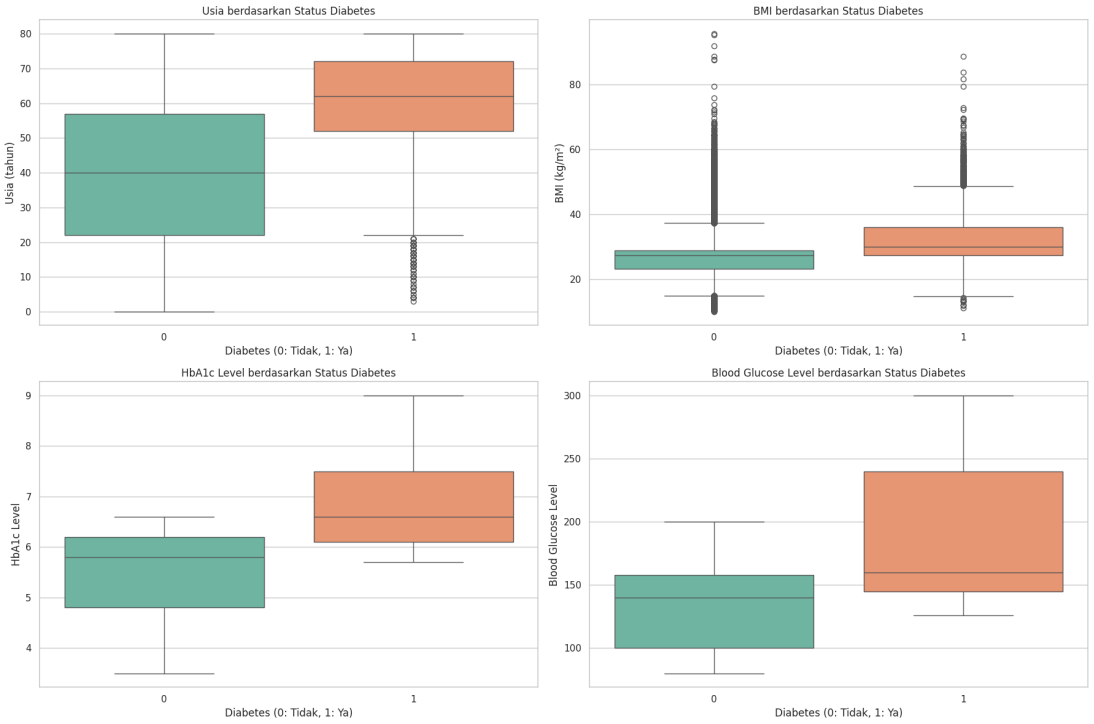
Grafik menunjukkan bahwa sebagian besar individu dalam dataset tidak memiliki riwayat merokok (**No Info** dan **never**) dengan jumlah tertinggi sekitar 35.000 data. Sementara kategori lain seperti **former**, **current**, **not current**, dan **ever** memiliki jumlah jauh lebih sedikit. Ini menunjukkan dominasi data non-perokok atau data tanpa informasi merokok yang jelas.

- Distribusi Fitur Numerik



Keempat grafik tersebut menggambarkan distribusi variabel numerik dalam dataset diabetes prediction. Distribusi usia menunjukkan bahwa data tersebar dari usia 0 hingga 80 tahun, dengan konsentrasi terbesar pada rentang usia 45 hingga 55 tahun, yang merupakan kelompok usia rawan terkena diabetes. Distribusi BMI memiliki bentuk sangat tajam di sekitar angka 25–30 kg/m², menunjukkan bahwa mayoritas individu dalam dataset berada dalam kategori kelebihan berat badan (overweight), yang merupakan faktor risiko diabetes. Sementara itu, distribusi kadar HbA1c memperlihatkan nilai-nilai diskrit yang dominan di sekitar angka 6%, yang merupakan ambang batas antara kondisi normal dan diabetes. Terakhir, distribusi kadar glukosa darah menunjukkan puncak di sekitar angka 150 mg/dL, yang juga merupakan nilai yang cukup tinggi dan mengindikasikan risiko diabetes. Secara keseluruhan, visualisasi ini memperlihatkan bahwa sebagian besar data memiliki karakteristik yang berhubungan erat dengan kondisi pradiabetes atau diabetes.

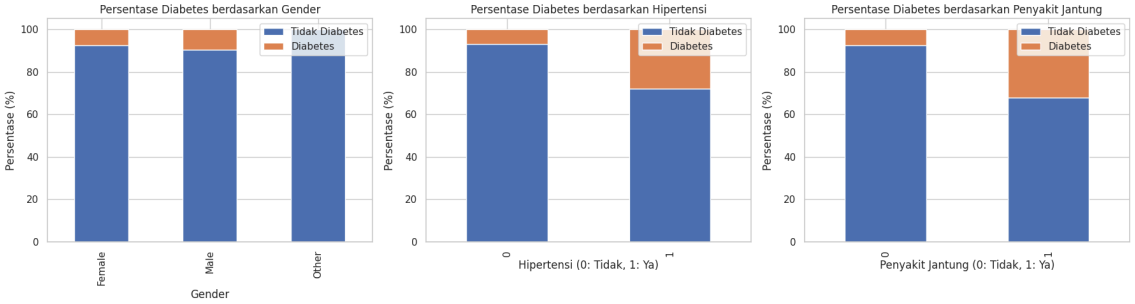
- Distribusi Fitur Numerik berdasarkan diabetes



Keempat boxplot ini membandingkan distribusi beberapa variabel numerik berdasarkan status diabetes (0 = Tidak, 1 = Ya). Dari plot usia, terlihat bahwa penderita diabetes umumnya berada pada rentang usia yang lebih tua, dengan median

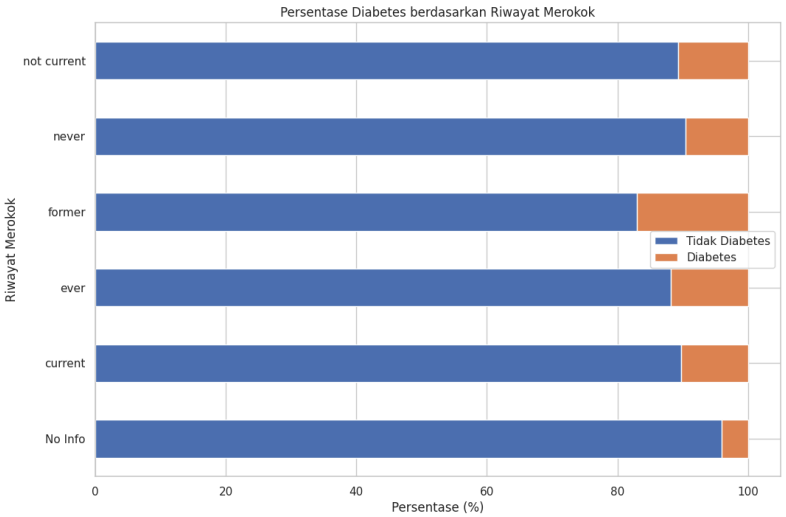
usia lebih tinggi dibandingkan yang tidak menderita diabetes. Pada plot BMI, meskipun distribusinya mirip, median BMI penderita diabetes sedikit lebih tinggi, dan terdapat lebih banyak nilai ekstrim (outlier) pada kelompok ini. Untuk kadar HbA1c, terdapat perbedaan mencolok, di mana individu dengan diabetes memiliki median HbA1c yang lebih tinggi, mengindikasikan bahwa HbA1c merupakan indikator yang kuat untuk diabetes. Begitu pula pada kadar glukosa darah, median dan rentang nilainya jauh lebih tinggi pada kelompok penderita diabetes, menunjukkan bahwa glukosa darah juga sangat berpengaruh terhadap status diabetes seseorang. Secara keseluruhan, visualisasi ini menunjukkan bahwa variabel usia, HbA1c, dan kadar glukosa darah sangat membedakan antara individu dengan dan tanpa diabetes.

- Korelasi antara fitur kategorikal dan target



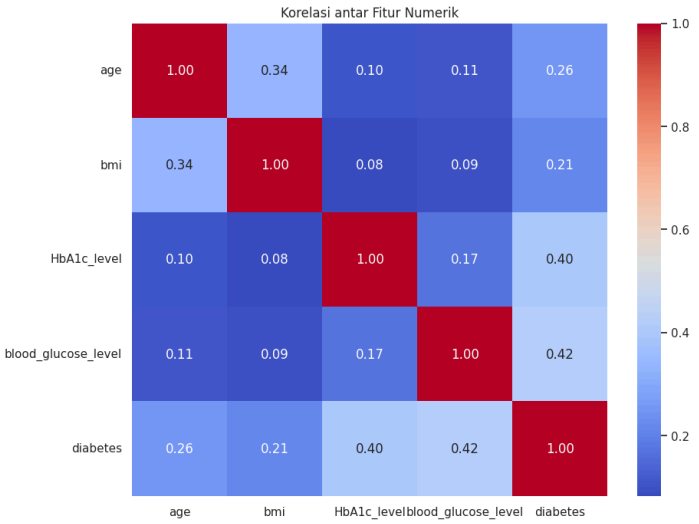
Gambar ini menunjukkan persentase penderita diabetes berdasarkan tiga variabel kategori: gender, hipertensi, dan penyakit jantung. Pada grafik pertama, persentase penderita diabetes hampir seragam antara laki-laki, perempuan, dan kategori lain, dengan sekitar 6–8% penderita diabetes di masing-masing kelompok gender. Grafik kedua menunjukkan bahwa proporsi penderita diabetes jauh lebih tinggi pada individu yang memiliki hipertensi dibandingkan yang tidak, menandakan adanya hubungan yang kuat antara hipertensi dan diabetes. Hal serupa terlihat pada grafik ketiga, di mana individu dengan riwayat penyakit jantung memiliki persentase diabetes yang lebih tinggi dibandingkan yang tidak memiliki penyakit jantung. Secara keseluruhan, hipertensi dan penyakit jantung tampaknya berkontribusi signifikan terhadap peningkatan risiko diabetes.

- Korelasi antara smoking history dan diabetes



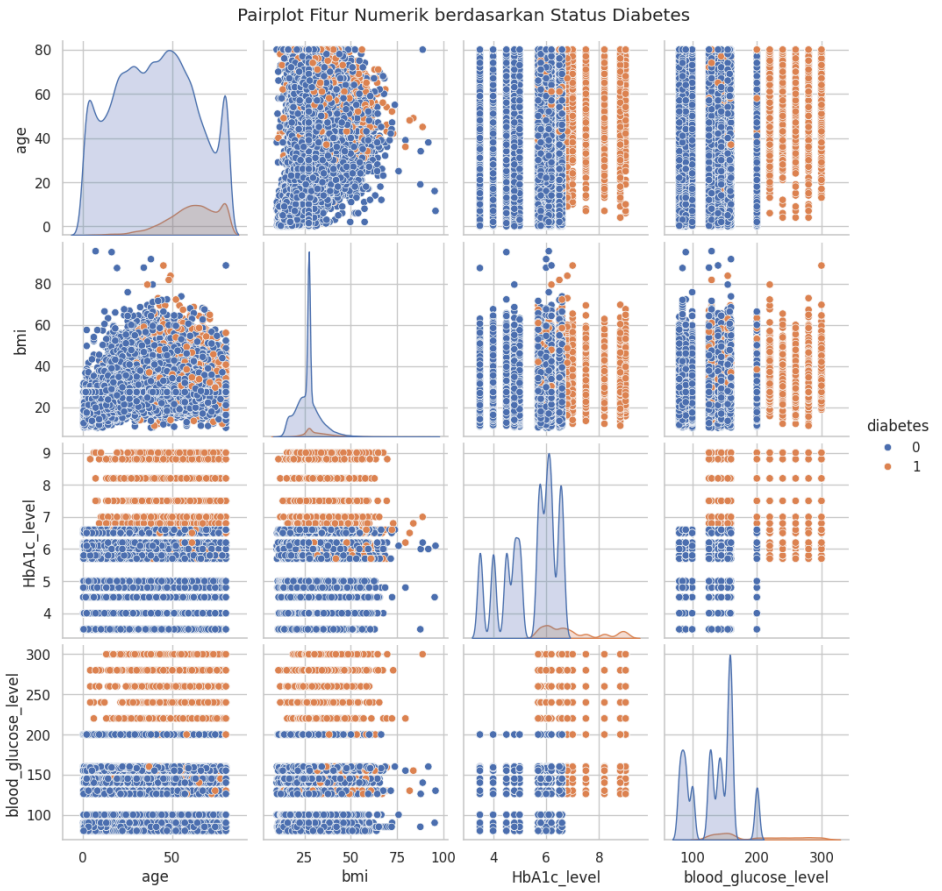
Grafik menunjukkan bahwa persentase penderita diabetes tertinggi terdapat pada kelompok mantan perokok (former), yang mengindikasikan bahwa individu yang pernah merokok dan telah berhenti memiliki risiko diabetes yang lebih besar dibandingkan kategori lainnya. Sementara itu, kategori seperti current (saat ini merokok), not current (pernah tetapi tidak merokok lagi), ever (pernah merokok), dan never (tidak pernah merokok) memiliki persentase penderita diabetes yang relatif lebih rendah dan hampir serupa. Kategori No Info menunjukkan persentase paling rendah, namun hal ini kemungkinan dipengaruhi oleh ketidaklengkapan data. Temuan ini menunjukkan bahwa riwayat merokok, terutama jika pernah merokok dalam jangka panjang, dapat berkontribusi terhadap risiko diabetes.

- Kolerasi antar fitur numerik



Heatmap tersebut menunjukkan korelasi antar fitur numerik dalam data, dengan fokus pada hubungan terhadap status diabetes. Fitur yang memiliki korelasi paling kuat dengan diabetes adalah *blood_glucose_level* (0.42) dan *HbA1c_level* (0.40), yang masuk dalam kategori korelasi sedang, menandakan bahwa kadar glukosa darah dan HbA1c cukup berkaitan dengan kemungkinan seseorang menderita diabetes. Usia (0.26) dan BMI (0.21) memiliki korelasi yang lebih lemah, meskipun tetap menunjukkan adanya hubungan. Korelasi antar fitur lainnya seperti antara usia dan BMI (0.34) juga tampak, tetapi tidak terlalu kuat. Secara keseluruhan, grafik ini membantu mengidentifikasi fitur-fitur yang relevan dalam memprediksi diabetes, di mana *blood_glucose_level* dan *HbA1c_level* menjadi yang paling berpengaruh.

- Pairplot untuk melihat hubungan antar fitur numerik



Gambar tersebut merupakan visualisasi pairplot yang menunjukkan hubungan antara fitur-fitur numerik dalam dataset berdasarkan status diabetes. Fitur yang ditampilkan meliputi usia (age), indeks massa tubuh (bmi), kadar HbA1c (HbA1c_level), dan kadar gula darah (blood_glucose_level), dengan warna biru mewakili individu non-diabetes dan warna oranye mewakili individu dengan diabetes. Dari grafik distribusi di sepanjang diagonal, terlihat bahwa individu dengan diabetes cenderung memiliki nilai HbA1c dan kadar gula darah yang lebih tinggi dibandingkan individu non-diabetes. Sementara itu, pada scatterplot antar fitur, hubungan antara HbA1c_level dan blood_glucose_level menunjukkan pola korelasi positif yang lebih jelas pada penderita diabetes. Namun, hubungan antara fitur lain seperti age dan bmi terhadap status diabetes tampak tidak terlalu mencolok. Secara keseluruhan, visualisasi ini memperlihatkan bahwa HbA1c_level dan blood_glucose_level merupakan fitur yang cukup kuat untuk membedakan individu dengan dan tanpa diabetes, sedangkan age dan bmi memiliki peran yang lebih lemah dalam hal ini.

D. Data Preparation

1. Handling Missing Values

   Setelah diperiksa, tidak ada nilai yang hilang dalam dataset ini

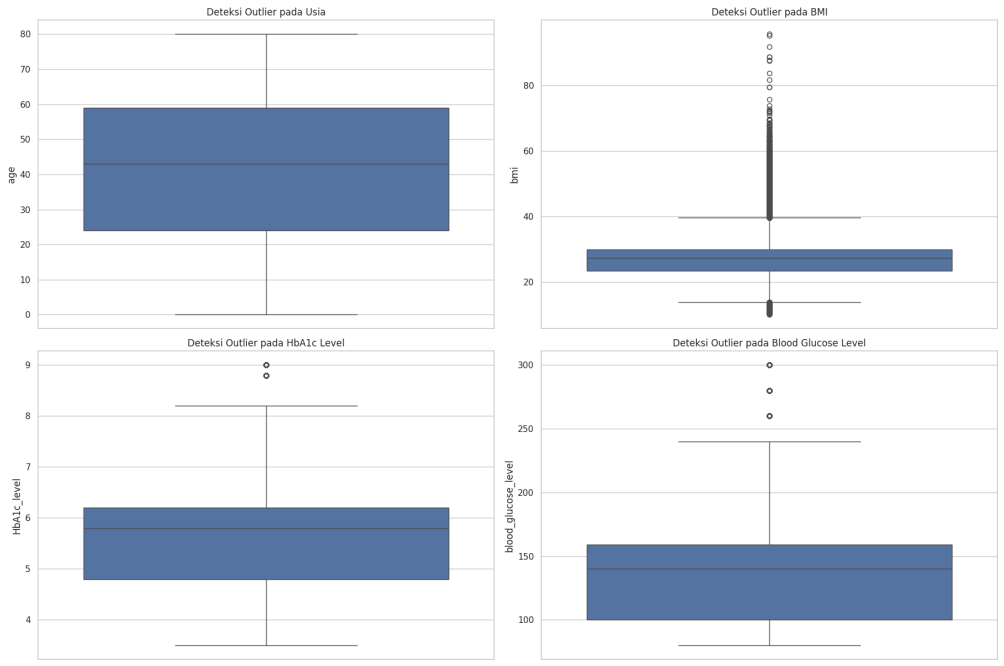
2. Handling Nilai Duplikat

   - Jumlah data sebelum menghapus duplikat : 100000
   - Jumlah data setelah menghapus duplikat : 96146
   - Jumlah duplikat yang dihapus : 3854

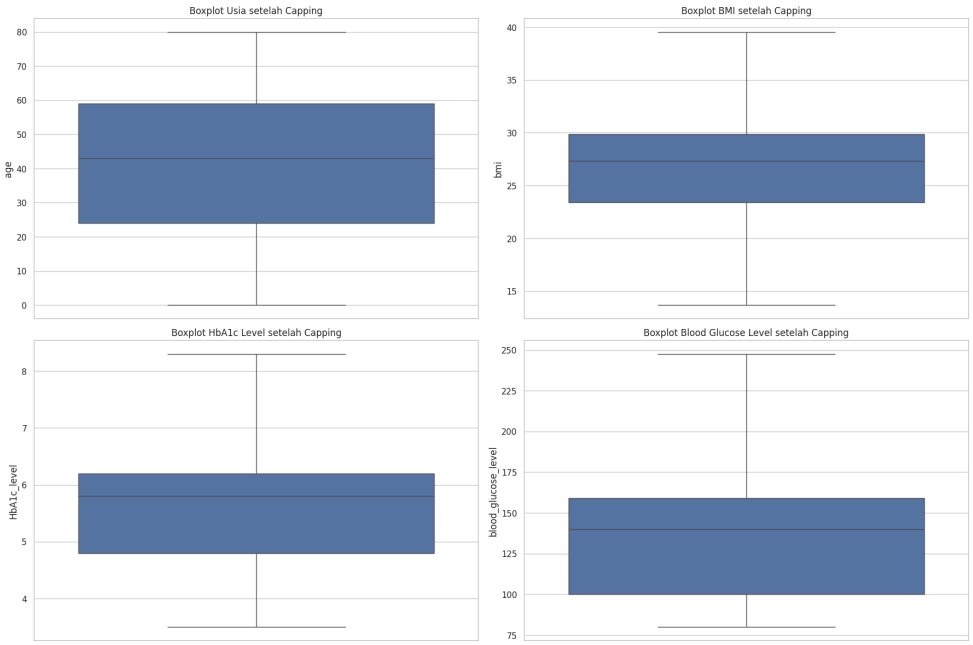
3. Handling Outlier

   Pada tahap ini, kita menangani outlier menggunakan metode capping, yaitu membatasi nilai outlier ke batas atas atau batas bawah yang telah ditentukan.

   1. Outlier yang ditangani pada age : 0
   2. Outlier yang ditangani pada BMI : 5354
   3. Outlier yang ditangani pada HbA1c_level : 1312
   4. Outlier yang ditangani pada blood_glucose_level : 2031
   5. Total outlier yang ditangani : 8697



Visualisasi boxplot sebelum outlier ditangani



Visualisasi boxplot setelah outlier ditangani

4. Encoding Fitur Kategorikal

   Kolom gender dan smoking_history dikonversi menggunakan teknik **One-Hot Encoding**.

```
# One-Hot Encoding untuk fitur kategorikal
df_encoded = pd.get_dummies(df, columns=['gender', 'smoking_history'], drop_first=True)

# Menampilkan hasil encoding
print("Hasil Encoding:")
display(df_encoded.head())
print(f"Jumlah kolom setelah encoding: {df_encoded.shape[1]}")
```

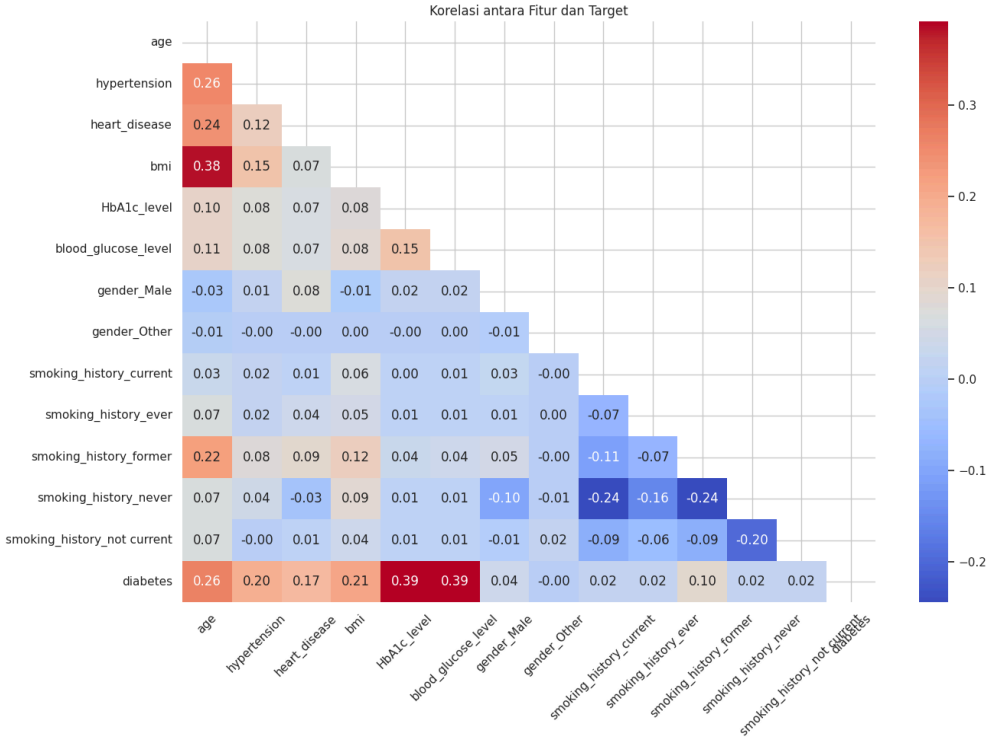
5. Normalisasi atau Standarisasi Data

   Kami menggunakan MinMaxScaler untuk menskala fitur numerik ke rentang 0-1. Hal ini membantu model machine learning konvergen lebih cepat dan memberikan bobot yang seimbang antar fitur.

```python
# Memisahkan fitur dan target
X = df_encoded.drop('diabetes', axis=1)
y = df_encoded['diabetes']

(variable) numeric_features: list[str]
numeric_features = ['age', 'bmi', 'HbA1c_level', 'blood_glucose_level']
categorical_features = [col for col in X.columns if col not in numeric_features]

# Standardisasi fitur numerik
scaler = MinMaxScaler()
X[numeric_features] = scaler.fit_transform(X[numeric_features])

# Menampilkan hasil standardisasi
print("Hasil Standardisasi:")
display(X.head())
```

6. Analisis Korelasi antara Variabel Independen dan Dependen

Pada tahap ini, kami menganalisis korelasi antara variabel independen dan variabel dependen untuk memahami hubungan antar variabel dan menentukan fitur-fitur yang paling berpengaruh terhadap target.



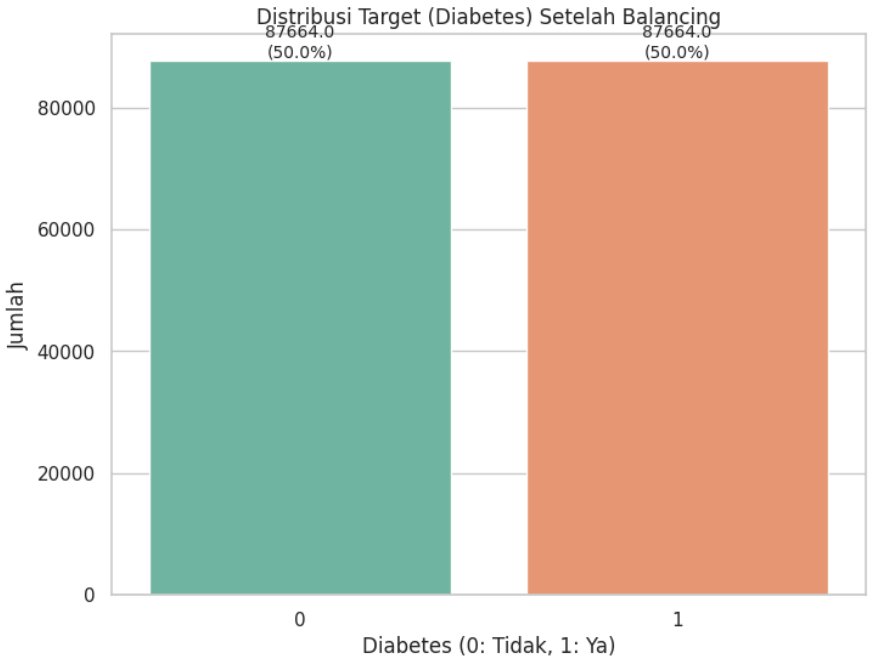
Hubungan antara variabel independen dan variabel dependen (diabetes) menunjukkan pola yang menarik. Kadar HbA1c dan kadar glukosa darah memiliki korelasi positif tertinggi dengan diabetes (keduanya 0.39), mengindikasikan bahwa kedua parameter laboratorium ini menjadi prediktor terkuat untuk diabetes, sesuai dengan pengetahuan medis bahwa keduanya merupakan indikator utama dalam diagnosis diabetes. BMI (0.21), usia (0.26), dan hipertensi (0.20) juga menunjukkan korelasi positif yang

cukup signifikan, menegaskan bahwa obesitas, penuaan, dan tekanan darah tinggi merupakan faktor risiko penting untuk diabetes. Penyakit jantung memiliki korelasi positif yang lebih lemah (0.17), menunjukkan hubungan yang ada namun tidak sekuat faktor-faktor sebelumnya. Sementara itu, faktor gender dan status merokok menunjukkan korelasi yang relatif lemah dengan diabetes (nilai korelasi mendekati nol), mengindikasikan bahwa kedua faktor tersebut mungkin tidak memiliki pengaruh signifikan terhadap risiko diabetes dalam dataset ini. Dari mantan perokok (0.10) menunjukkan korelasi positif lemah dibandingkan dengan kategori merokok lainnya. Temuan ini menyarankan bahwa model prediktif untuk diabetes sebaiknya memberikan perhatian lebih pada variabel biomedis seperti kadar glukosa darah, HbA1c, BMI, dan faktor demografis seperti usia, sambil tetap mempertimbangkan faktor-faktor lain dengan korelasi yang lebih lemah.

7. Menangani Ketidakseimbangan Kelas

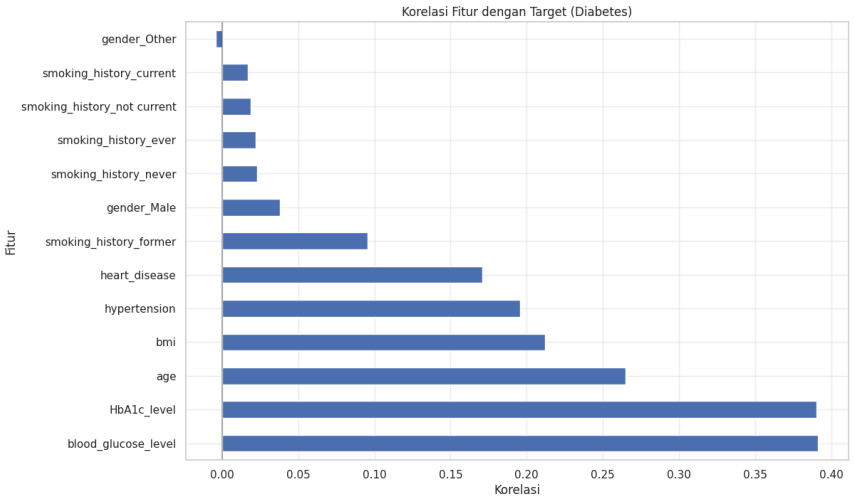
Pada tahap ini, kami menangani ketidakseimbangan kelas dalam dataset menggunakan teknik SMOTE (Synthetic Minority Over-sampling Technique). SMOTE digunakan untuk menyeimbangkan kelas dalam dataset. Sebelum balancing, kelas 0 (tidak diabetes) memiliki 87.664 data, sedangkan kelas 1 (diabetes) hanya memiliki 8.482 data. Setelah balancing dengan SMOTE, kedua kelas memiliki jumlah data yang sama, yaitu 87.664 data.

- Distribusi Target (Diabetes) setelah Balancing



Visualisasi distribusi kelas setelah balancing menunjukkan bahwa kedua kelas sekarang memiliki jumlah data yang sama.

- Korelasi variabel dengan target



Visualisasi bar chart menunjukkan korelasi berbagai fitur dengan diabetes. Kadar glukosa darah dan HbA1c memiliki korelasi tertinggi (±0.39), menandakan peran krusialnya sebagai prediktor diabetes. Usia menempati posisi ketiga (±0.26), diikuti BMI (±0.21) dan hipertensi (±0.18), yang menegaskan bahwa faktor usia, berat badan, dan tekanan darah tinggi berkontribusi signifikan terhadap risiko diabetes. Penyakit jantung menunjukkan korelasi sedang (±0.16). Status "mantan perokok" memiliki korelasi lebih tinggi (±0.10) dibanding kategori merokok lainnya. Gender laki-laki dan kategori merokok lainnya menunjukkan korelasi rendah (<0.05), sementara gender "Other" memiliki korelasi terendah, hampir mendekati nol. Visualisasi ini mengonfirmasi bahwa faktor biomedis dan usia lebih kuat berkorelasi dengan diabetes dibandingkan faktor demografis dan perilaku merokok.

E. Implementasi Model

1. Train - Test Split

   Pada tahap ini, kita membagi dataset menjadi data latih dan data uji dengan perbandingan 80:20.

```
# Memisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print(f"Jumlah data latih: {X_train.shape[0]}")
print(f"Jumlah data uji: {X_test.shape[0]}")

# Memeriksa distribusi target pada data latih dan uji
print("\nDistribusi Target pada Data Latih:")
display(pd.Series(y_train).value_counts(normalize=True))

print("\nDistribusi Target pada Data Uji:")
display(pd.Series(y_test).value_counts(normalize=True))
```

Data dibagi menjadi 76.916 data latih dan 19.230 data uji. Parameter `stratify=y` memastikan bahwa distribusi kelas dalam data latih dan data uji sama dengan distribusi kelas dalam dataset asli.

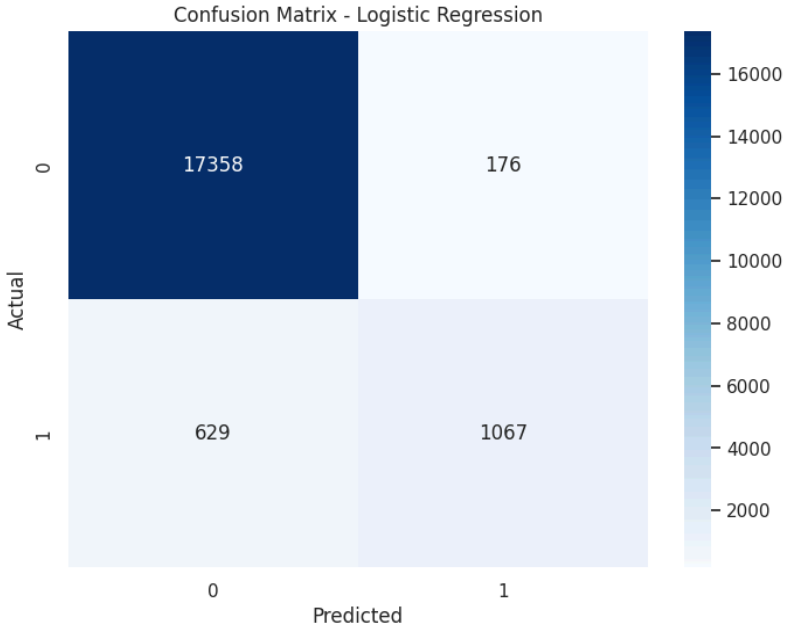
2. **Membangun Model Logistic Regression**

```
# Membangun model Logistic Regression
logistic_regression = LogisticRegression(random_state=42, max_iter=1000)
logistic_regression.fit(X_train, y_train)

# Prediksi pada data uji
y_pred_lr = logistic_regression.predict(X_test)
y_prob_lr = logistic_regression.predict_proba(X_test)[:, 1]

# Evaluasi model
print("Evaluasi Model Logistic Regression:")
print(f"Accuracy: {accuracy_score(y_test, y_pred_lr):.4f}")
print(f"Precision: {precision_score(y_test, y_pred_lr):.4f}")
print(f"Recall: {recall_score(y_test, y_pred_lr):.4f}")
print(f"F1 Score: {f1_score(y_test, y_pred_lr):.4f}")

# Confusion Matrix
cm_lr = confusion_matrix(y_test, y_pred_lr)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_lr, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Logistic Regression')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()

# ROC Curve
fpr_lr, tpr_lr, _ = roc_curve(y_test, y_prob_lr)
roc_auc_lr = auc(fpr_lr, tpr_lr)

plt.figure(figsize=(8, 6))
plt.plot(fpr_lr, tpr_lr, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc_lr:.4f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - Logistic Regression')
plt.legend(loc="lower right")
plt.show()
```

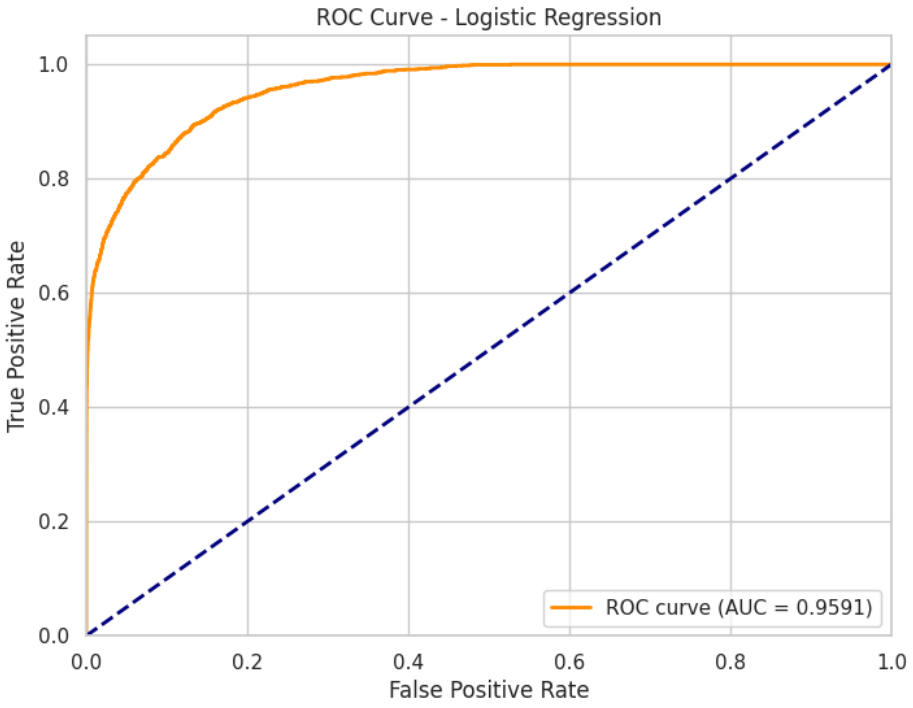
Evaluasi Model Logistic Regression :

- Accuracy : 0.9581
- Precision : 0.8584
- Recall : 0.6291
- F1 Score : 0.7261

Confusion Matrix - Logistic Regression



ROC Curve Logistic Regression



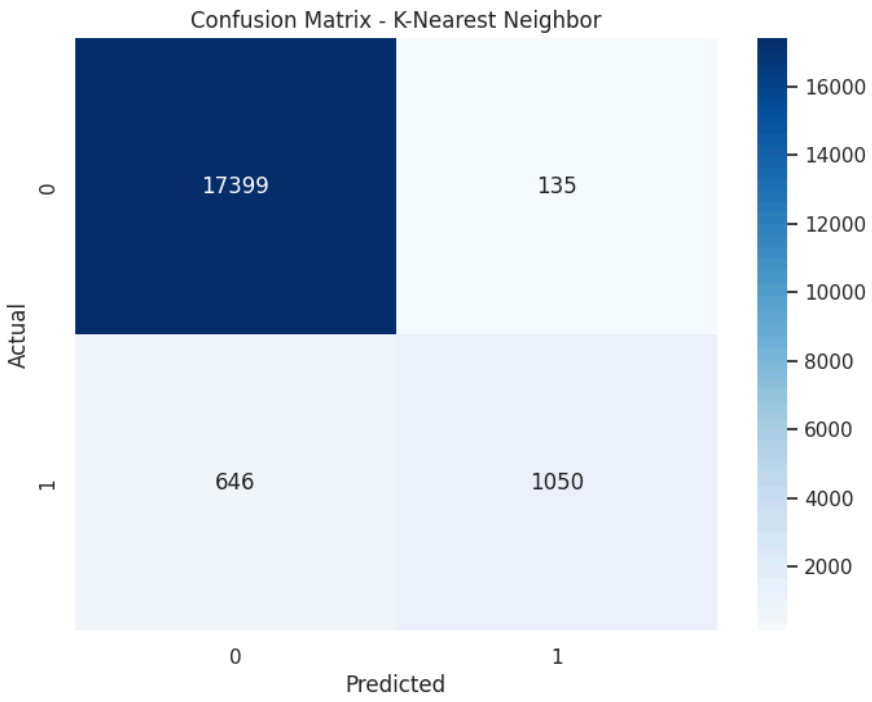
3. Membangun Model K-Nearest Neighbor

```python
# Membangun model K-Nearest Neighbor
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)

# Prediksi pada data uji
y_pred_knn = knn.predict(X_test)
y_prob_knn = knn.predict_proba(X_test)[:, 1]

# Evaluasi model
print("Evaluasi Model K-Nearest Neighbor:")
print(f"Accuracy: {accuracy_score(y_test, y_pred_knn):.4f}")
print(f"Precision: {precision_score(y_test, y_pred_knn):.4f}")
print(f"Recall: {recall_score(y_test, y_pred_knn):.4f}")
print(f"F1 Score: {f1_score(y_test, y_pred_knn):.4f}")

# Confusion Matrix
cm_knn = confusion_matrix(y_test, y_pred_knn)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_knn, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - K-Nearest Neighbor')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()

# ROC Curve
fpr_knn, tpr_knn, _ = roc_curve(y_test, y_prob_knn)
roc_auc_knn = auc(fpr_knn, tpr_knn)

plt.figure(figsize=(8, 6))
plt.plot(fpr_knn, tpr_knn, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc_knn:.4f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - K-Nearest Neighbor')
plt.legend(loc="lower right")
plt.show()
```

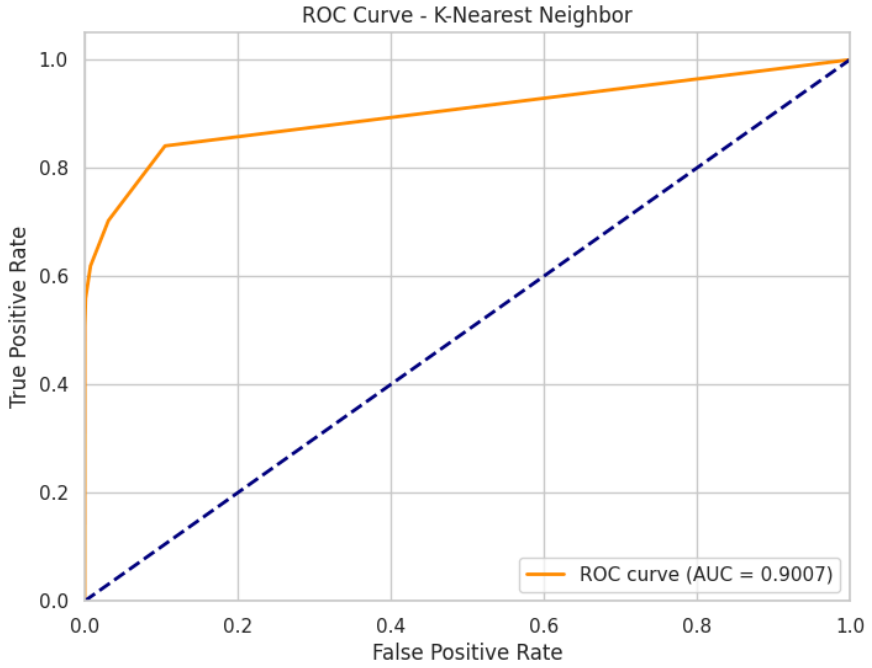
Evaluasi Model K-Nearest Neighbor :

- Accuracy : 0.9594
- Precision : 0.8861
- Recall : 0.6191
- F1 Score : 0.7289

Confusion Matrix K-Nearest Neighbor



ROC Curve K-Nearest Neighbor



4. Membangun Model Naive Bayes

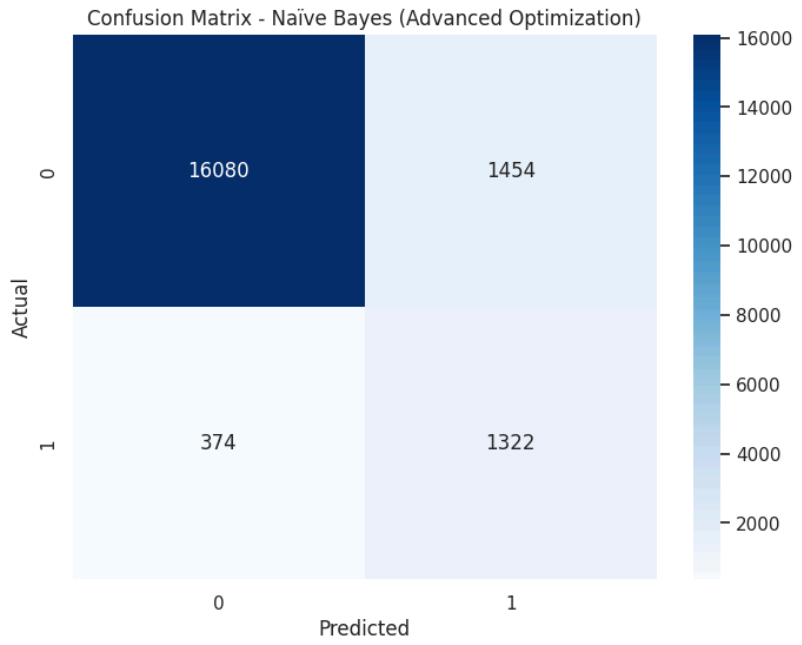
Membangun model prediksi Naive Bayes yang telah dioptimalkan melalui serangkaian tahapan preprocessing dan tuning. Pertama, data disalin dan fitur kategorikal seperti *gender* dan *smoking_history* diencoding menggunakan one-hot encoding. Kemudian, beberapa fitur numerik penting seperti *HbA1c_level*, *blood_glucose_level*, *bmi*, dan *age* dibagi ke dalam kategori (binning) berdasarkan

pengetahuan domain medis, dan hasil binning ini juga diubah menjadi one-hot encoding. Selanjutnya, fitur numerik dinormalisasi menggunakan Min-Max Scaling, dan data dibagi menjadi data latih dan uji dengan stratifikasi untuk menjaga proporsi kelas. Untuk meningkatkan kinerja Naive Bayes, dilakukan seleksi fitur menggunakan metode chi-squared untuk memilih fitur paling informatif, kemudian dilakukan resampling menggunakan kombinasi SMOTE dan Tomek links guna menyeimbangkan distribusi kelas. Model kemudian diuji menggunakan dua varian Naive Bayes—*GaussianNB* dan *ComplementNB*—yang dioptimalkan melalui GridSearchCV untuk menemukan parameter terbaik berdasarkan skor F1. Model terbaik dari keduanya digabung dalam ensemble dengan VotingClassifier untuk memanfaatkan kekuatan keduanya. Setelah dilatih, model dievaluasi melalui metrik seperti akurasi, presisi, recall, F1-score, confusion matrix, dan kurva ROC untuk melihat performanya dalam mengklasifikasikan diabetes secara lebih akurat dan andal.

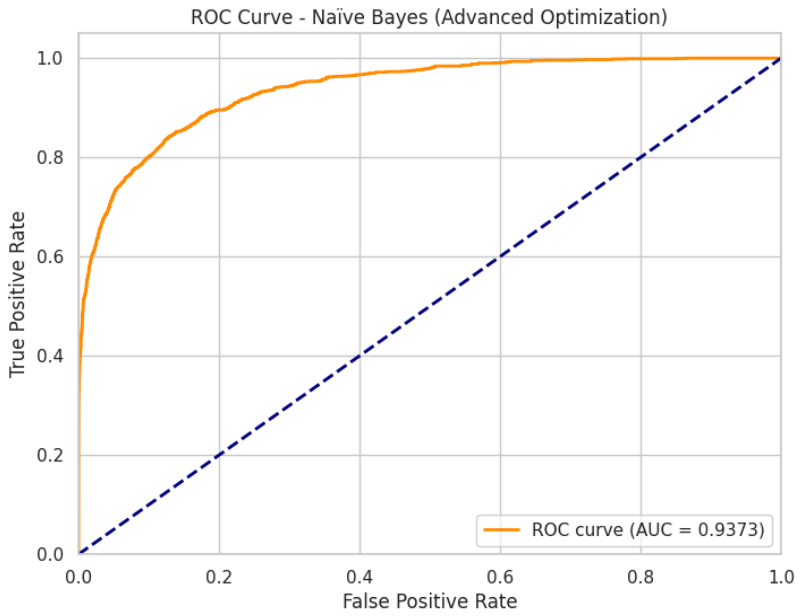
Evaluasi Model Naive Bayes :

- Accuracy : 0.9040
- Precision : 0.4762
- Recall : 0.7795
- F1 Score : 0.5912

Confusion Matrix Naive Bayes



ROC Curve Naive Bayes



5. Membangun Model Decision Tree

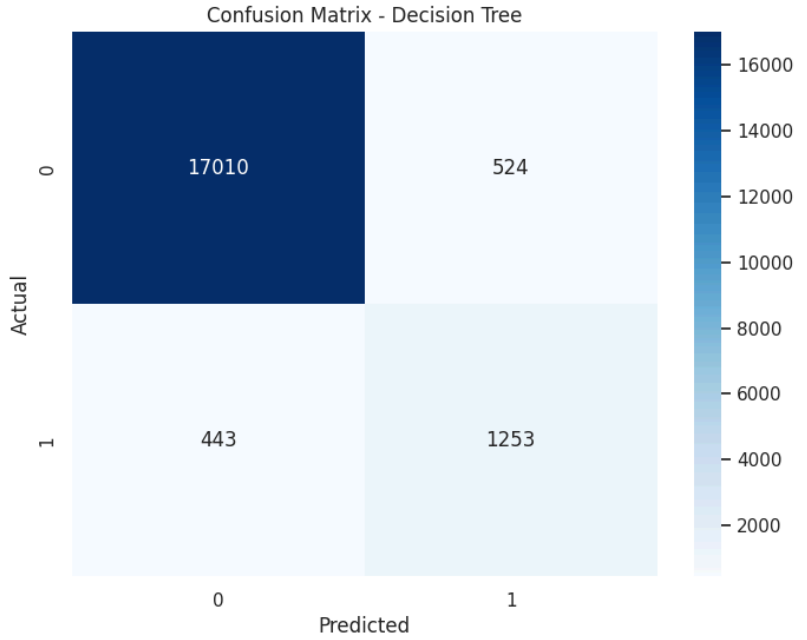
```
# Membangun model Decision Tree
decision_tree = DecisionTreeClassifier(random_state=42)
decision_tree.fit(X_train, y_train)

# Prediksi pada data uji
y_pred_dt = decision_tree.predict(X_test)
y_prob_dt = decision_tree.predict_proba(X_test)[:, 1]

# Evaluasi model
print("Evaluasi Model Decision Tree:")
print(f"Accuracy: {accuracy_score(y_test, y_pred_dt):.4f}")
print(f"Precision: {precision_score(y_test, y_pred_dt):.4f}")
print(f"Recall: {recall_score(y_test, y_pred_dt):.4f}")
print(f"F1 Score: {f1_score(y_test, y_pred_dt):.4f}")

# Confusion Matrix
cm_dt = confusion_matrix(y_test, y_pred_dt)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_dt, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Decision Tree')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()

# ROC Curve
fpr_dt, tpr_dt, _ = roc_curve(y_test, y_prob_dt)
roc_auc_dt = auc(fpr_dt, tpr_dt)

plt.figure(figsize=(8, 6))
plt.plot(fpr_dt, tpr_dt, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc_dt:.4f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - Decision Tree')
plt.legend(loc="lower right")
plt.show()
```

Evaluasi Model Decision Tree :

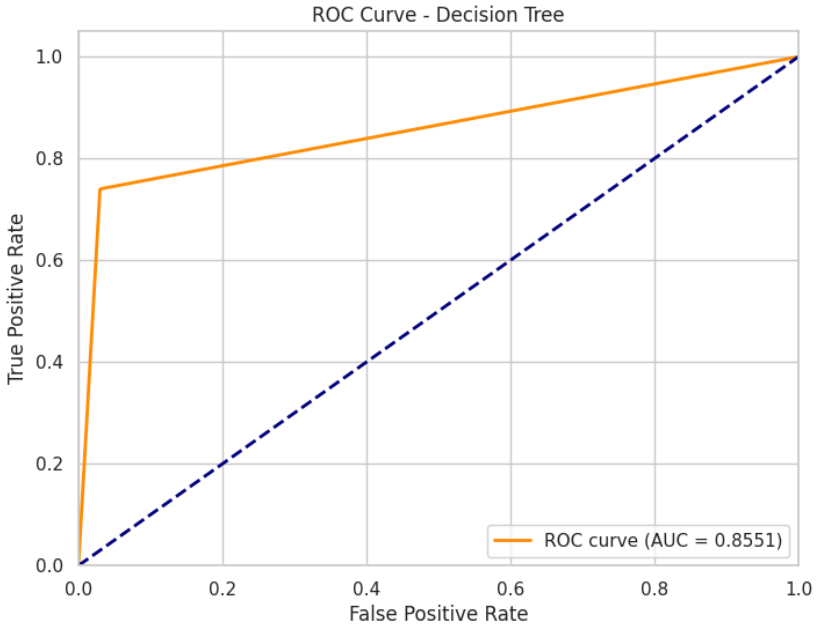
- Accuracy : 0.9497
- Precision : 0.7051
- Recall : 0.7388

- F1 Score : 0.7216

Confusion Matrix Decision Tree

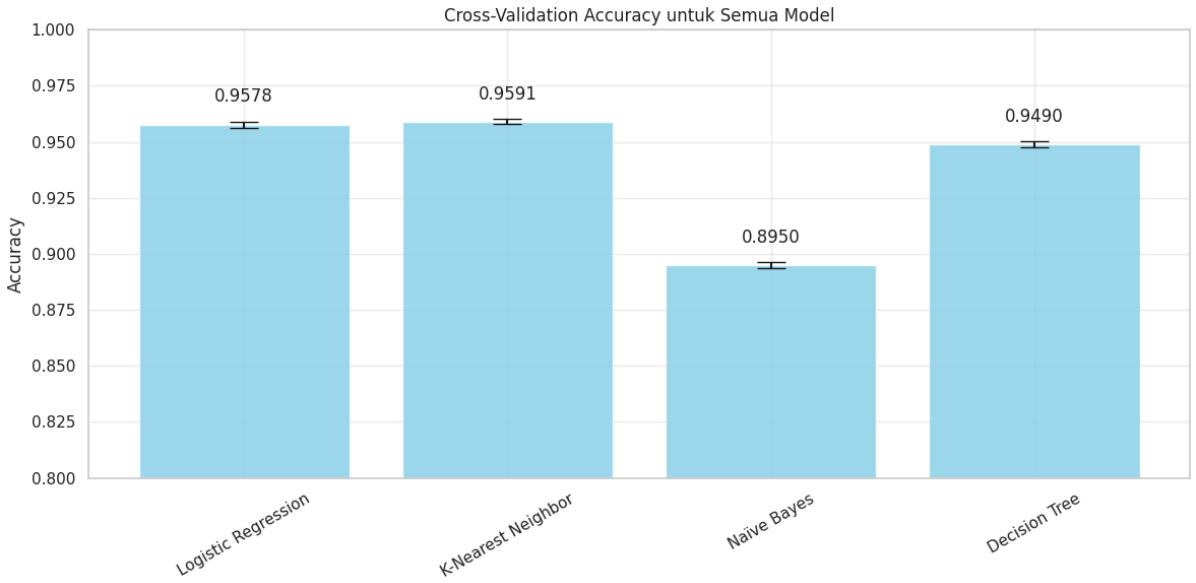


ROC Curve Decision Tree



6. Cross-Validation

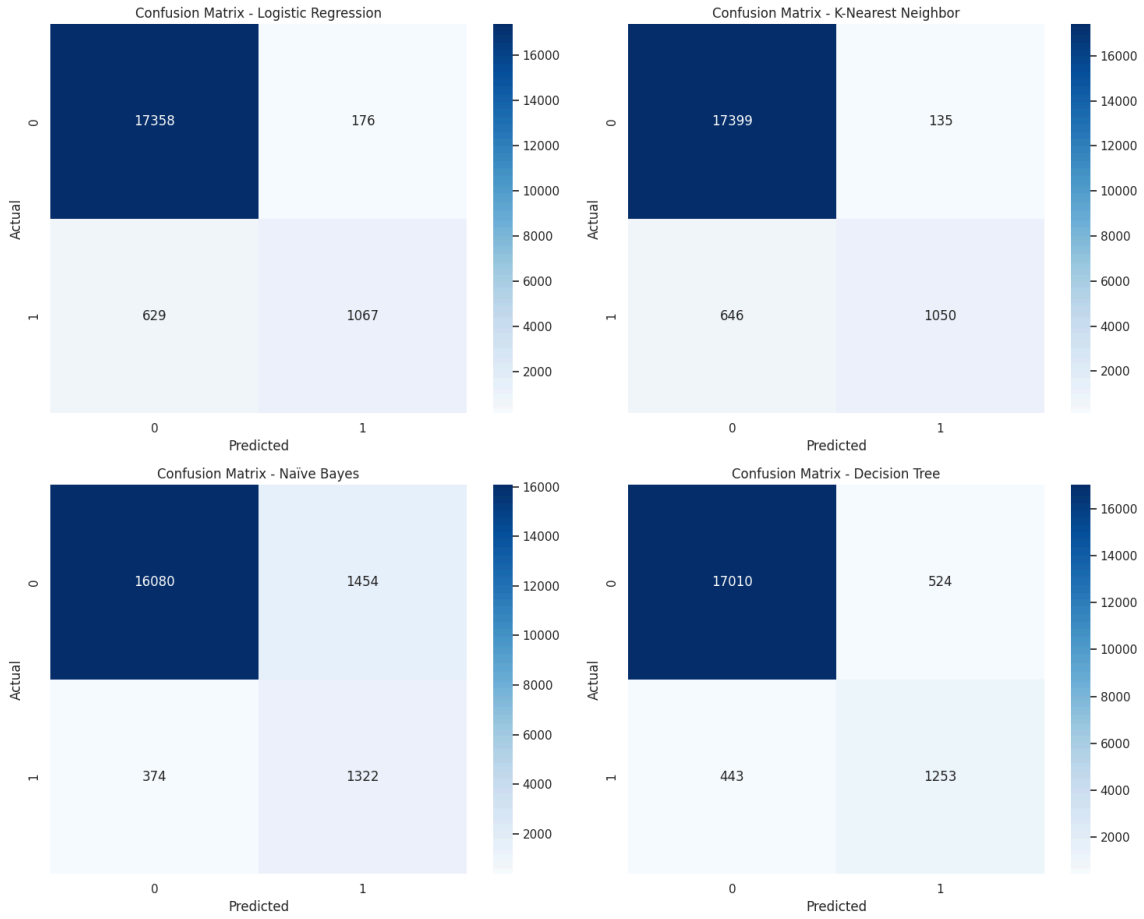
Cross-validation digunakan untuk mengevaluasi performa model dengan lebih reliabel, dengan membagi data menjadi beberapa fold dan melatih model pada kombinasi fold yang berbeda.




```
Model: Logistic Regression
Cross-Validation Accuracy: 0.9578 ± 0.0014
All Scores: [0.95585023 0.95902023 0.956472   0.95818815
0.95922825]
--------------------------------------------------
Model: K-Nearest Neighbor
Cross-Validation Accuracy: 0.9591 ± 0.0013
All Scores: [0.95855434 0.95886422 0.95720006 0.95969629
0.96110042]
--------------------------------------------------
Model: Naïve Bayes
Cross-Validation Accuracy: 0.8950 ± 0.0012
All Scores: [0.89360374 0.89500234 0.89588642 0.8967705
0.89385823]
--------------------------------------------------
Model: Decision Tree
Cross-Validation Accuracy: 0.9490 ± 0.0013
All Scores: [0.94706188 0.94804722 0.94945135 0.95054345
0.9500234 ]
```

F. Evaluasi Model

1. Confusion Matrix



Visualisasi confusion matrix untuk semua model membantu kita membandingkan performa model dalam hal True Positive, True Negative, False Positive, dan False Negative. Dari confusion matrix, kita dapat melihat bahwa:

1. Model Logistic Regression memiliki True Negative yang tinggi tetapi False Negative yang juga relatif tinggi.
2. Model K-Nearest Neighbor memiliki pola yang mirip dengan Logistic Regression.
3. Model Naïve Bayes memiliki False Positive yang tinggi, tetapi False Negative yang rendah.
4. Model Decision Tree memiliki keseimbangan yang cukup baik antara False Positive dan False Negative.

2. Classification Report

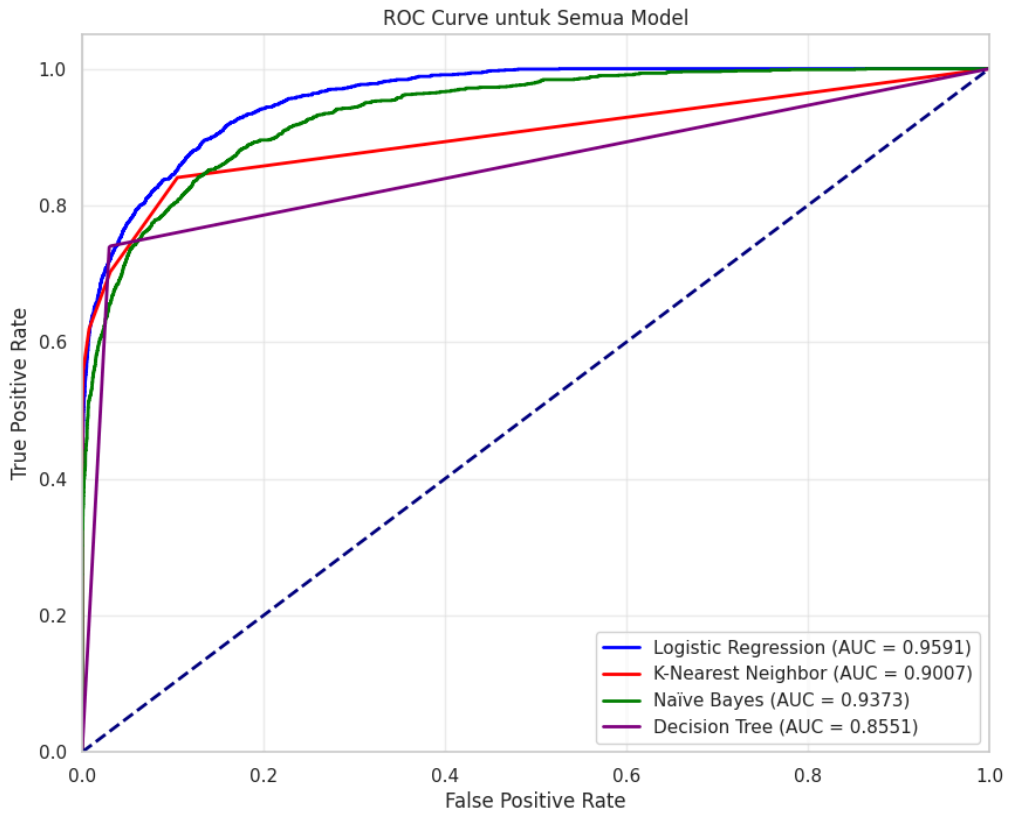
```
Classification Report - Logistic Regression:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     17534
           1       0.86      0.63      0.73      1696
```

```
    accuracy                           0.96     19230
   macro avg       0.91      0.81      0.85     19230
weighted avg       0.96      0.96      0.96     19230

--------------------------------------------------------------------------
----
Classification Report - K-Nearest Neighbor:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     17534
           1       0.89      0.62      0.73      1696

    accuracy                           0.96     19230
   macro avg       0.93      0.81      0.85     19230
weighted avg       0.96      0.96      0.96     19230

--------------------------------------------------------------------------
----
Classification Report - Naïve Bayes:
              precision    recall  f1-score   support

...
  macro avg       0.84      0.85      0.85     19230
weighted avg       0.95      0.95      0.95     19230
```

Classification report memberikan informasi detail tentang performa model untuk setiap kelas. Dari classification report, kita dapat melihat bahwa:

1. Model Logistic Regression memiliki precision yang tinggi tetapi recall yang rendah untuk kelas positive (diabetes).
2. Model K-Nearest Neighbor memiliki precision yang sedikit lebih tinggi dari Logistic Regression tetapi recall yang sedikit lebih rendah.
3. Model Naïve Bayes memiliki precision yang rendah tetapi recall yang tinggi untuk kelas positive.
4. Model Decision Tree memiliki keseimbangan yang baik antara precision dan recall.

3. ROC Curve



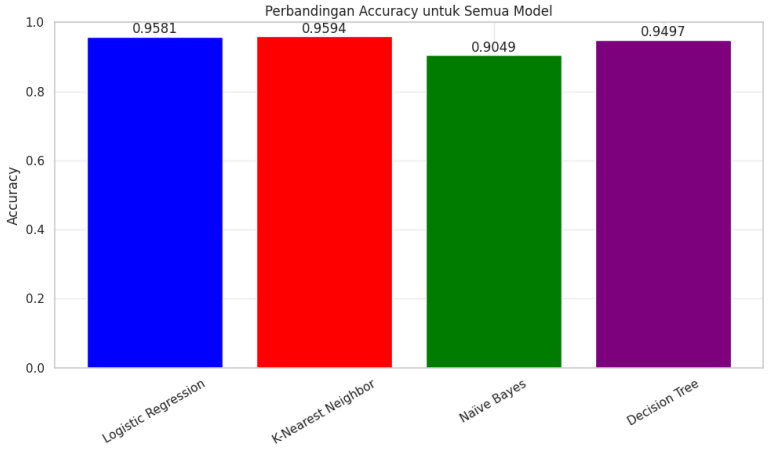
ROC curve menunjukkan trade-off antara True Positive Rate dan False Positive Rate untuk berbagai threshold. Model yang ideal akan memiliki kurva yang mendekati sudut kiri atas, yang menunjukkan True Positive Rate yang tinggi dan False Positive Rate yang rendah. Area Under the Curve (AUC) adalah ukuran keseluruhan performa model, di mana nilai yang lebih tinggi menunjukkan model yang lebih baik.

Dari ROC curve, kita dapat melihat bahwa:

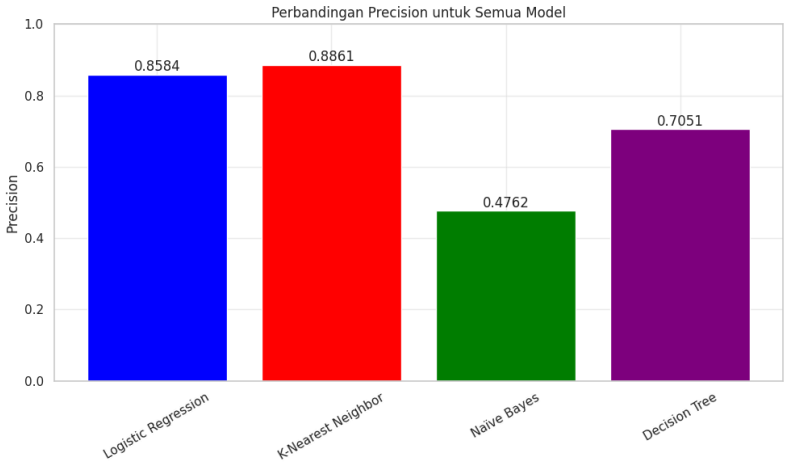
1. Model Logistic Regression memiliki AUC tertinggi (0.9591).
2. Model Naïve Bayes memiliki AUC kedua tertinggi (0.9373).
3. Model K-Nearest Neighbor memiliki AUC ketiga tertinggi (0.9007).
4. Model Decision Tree memiliki AUC terendah (0.8551).

4. Perbandingan Matrix Evaluasi

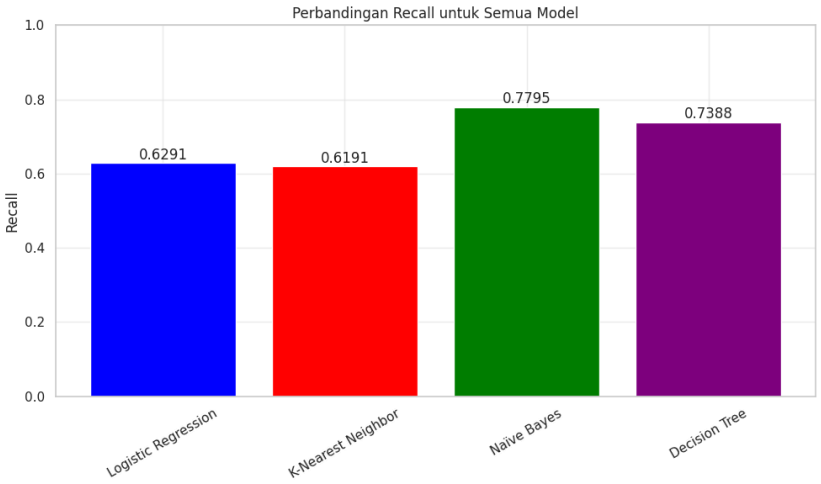
- **Perbandingan Accuracy untuk Semua Model**



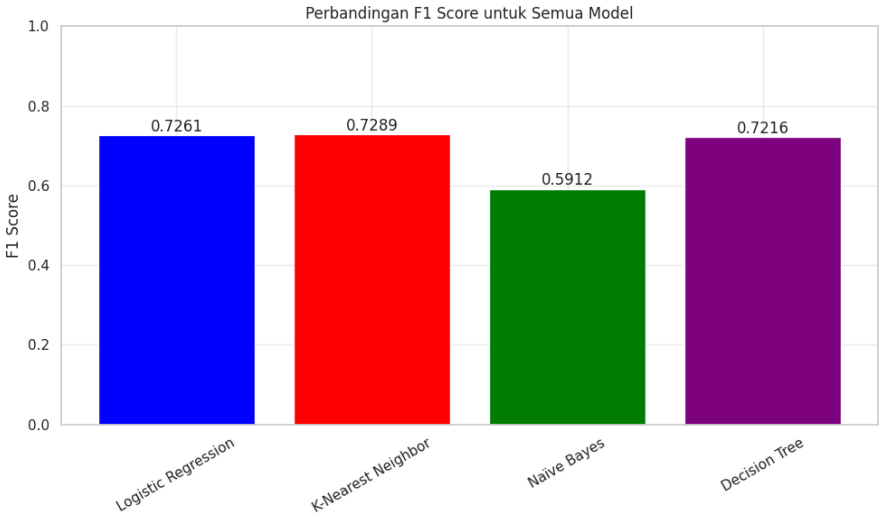
- **Perbandingan Precision untuk Semua Model**



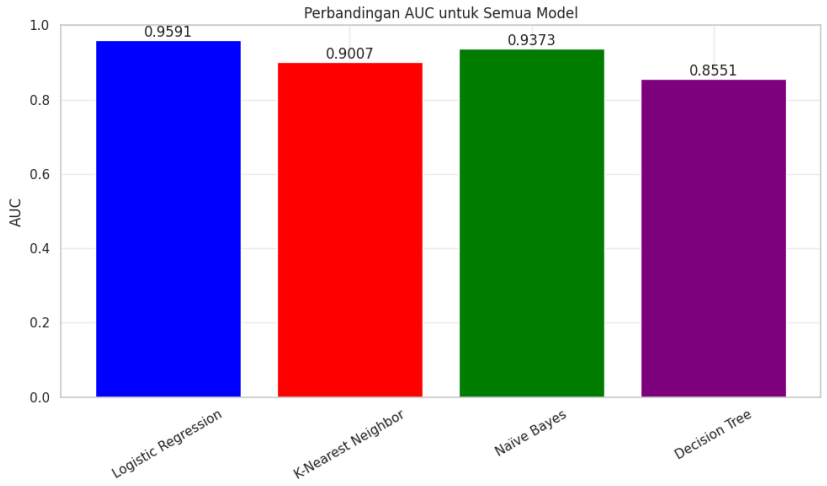
- **Perbandingan Recall untuk Semua Model**



- **Perbandingan F1 Score untuk Semua Model**



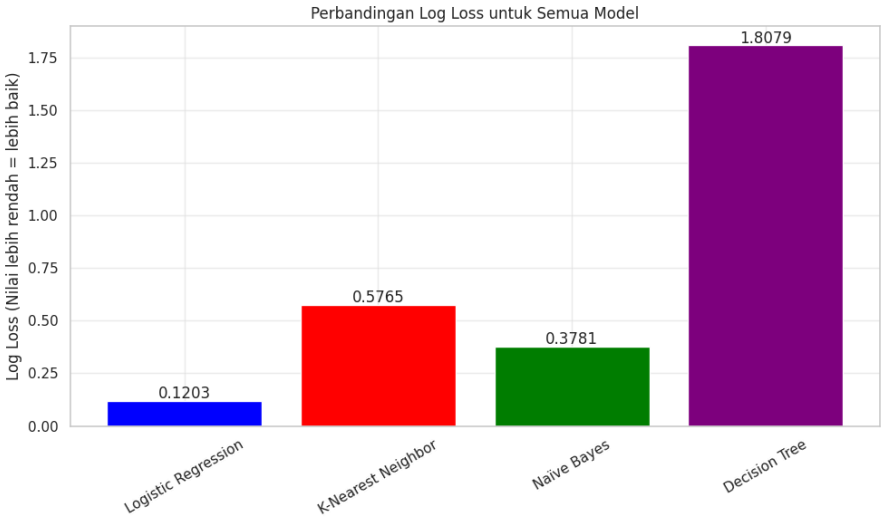
- **Perbandingan AUC untuk Semua Model**



Visualisasi metrik evaluasi membantu kita membandingkan performa model untuk berbagai metrik. Dari visualisasi ini, kita dapat melihat bahwa:

1. Model K-Nearest Neighbor memiliki akurasi tertinggi.
2. Model K-Nearest Neighbor juga memiliki precision tertinggi.
3. Model Naïve Bayes memiliki recall tertinggi.
4. Model K-Nearest Neighbor memiliki F1 score tertinggi.
5. Model Logistic Regression memiliki AUC tertinggi.

5. Loss Value



Log loss memberikan ukuran seberapa baik model dalam memprediksi probabilitas kelas. Nilai log loss yang lebih rendah menunjukkan model yang lebih baik. Dari visualisasi log loss, kita dapat melihat bahwa:

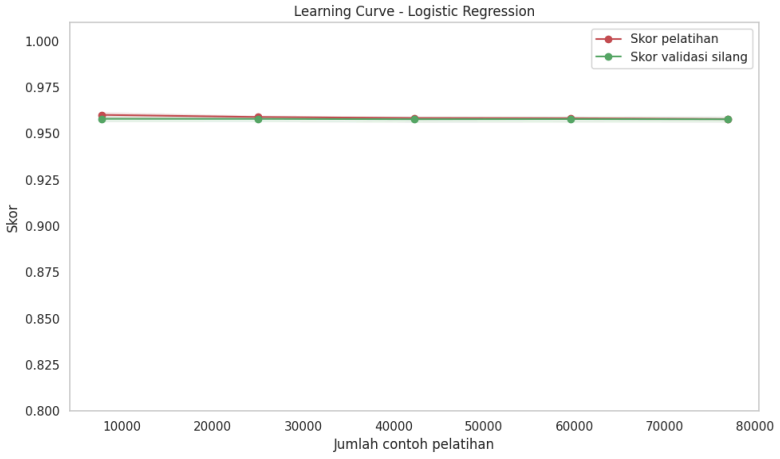
1. Model Logistic Regression memiliki log loss terendah (0.1203).
2. Model Naïve Bayes memiliki log loss kedua terendah (0.3781).
3. Model K-Nearest Neighbor memiliki log loss ketiga terendah (0.5765).
4. Model Decision Tree memiliki log loss tertinggi (1.8079).

G. Analisis Hasil

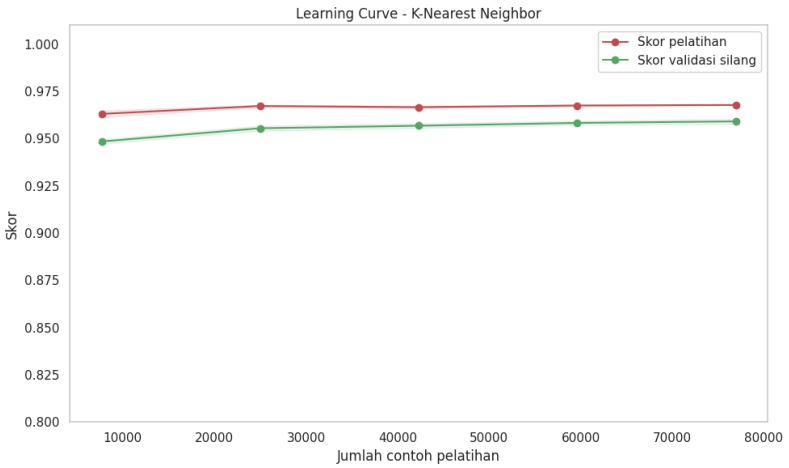
1. Grafik Fit Untuk Beberapa Algoritma

   Learning curve membantu kita memahami bagaimana performa model berubah seiring dengan peningkatan jumlah data latih. Learning curve juga membantu kita mendeteksi masalah overfitting dan underfitting.

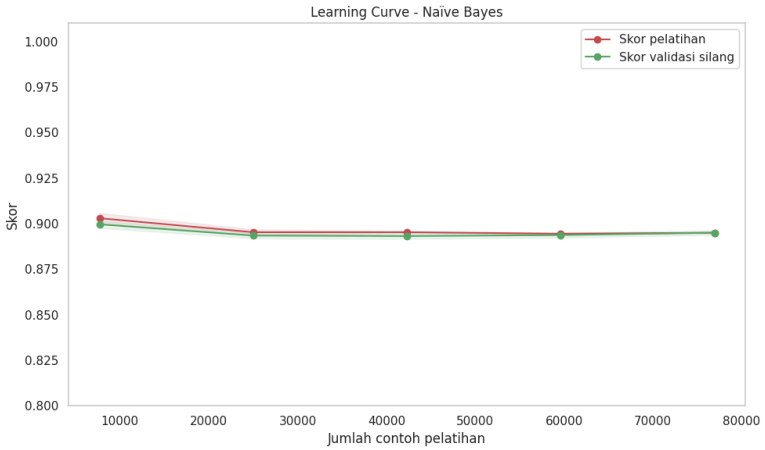
   - Learning Curve Logistic Regression



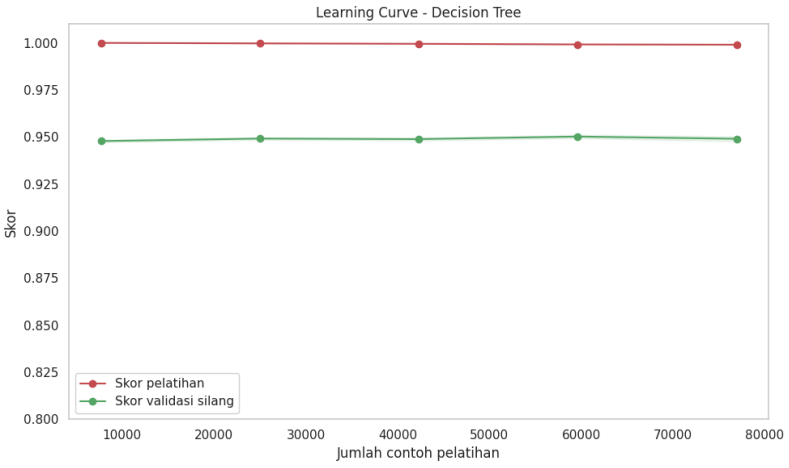
   - Learning Curve K-Nearest Neighbor



- Learning Curve Naive Bayes



- Decision Tree



Learning curve menunjukkan bagaimana performa model berubah seiring dengan peningkatan jumlah data latih. Jika skor validasi silang (garis hijau) masih meningkat dengan peningkatan jumlah data latih, maka model dapat memperoleh manfaat dari lebih banyak data. Jika terdapat celah besar antara skor pelatihan dan skor validasi silang, maka model mungkin mengalami overfitting.

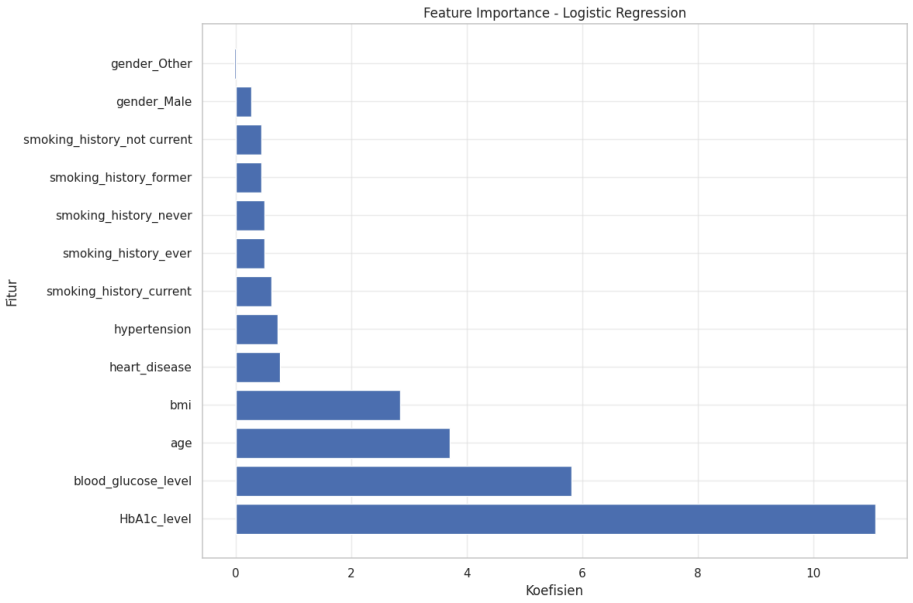
Dari learning curve, kita dapat melihat bahwa:

1. Model Logistic Regression memiliki celah kecil antara skor pelatihan dan skor validasi silang, menunjukkan bahwa model ini memiliki bias dan varians yang seimbang.
2. Model K-Nearest Neighbor memiliki celah yang sedikit lebih besar, menunjukkan bahwa model ini mungkin sedikit overfitting.
3. Model Naïve Bayes memiliki celah yang cukup besar, menunjukkan bahwa model ini mungkin underfitting.
4. Model Decision Tree memiliki celah yang paling besar, menunjukkan bahwa model ini mungkin overfitting.

2. Feature Importance

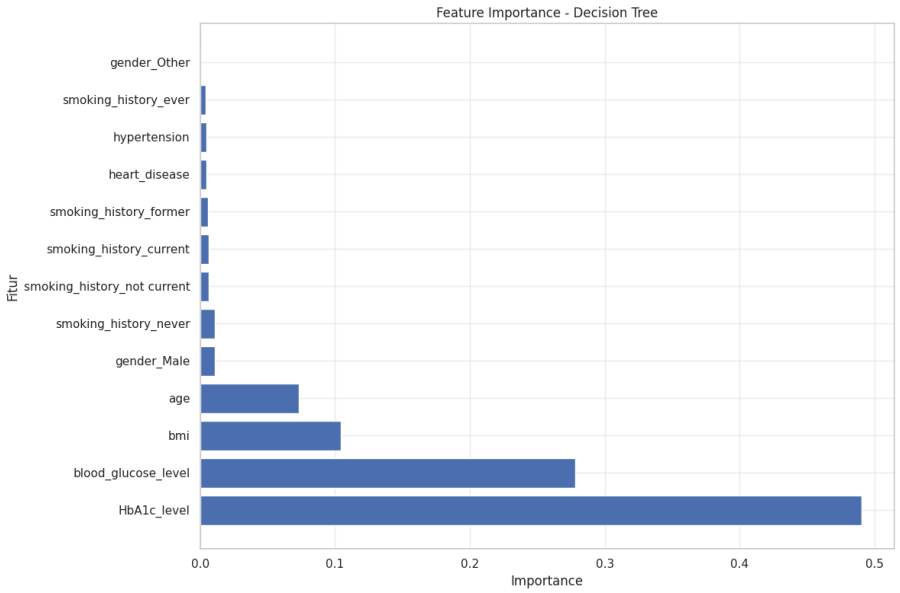
Feature importance menunjukkan seberapa penting setiap fitur dalam memprediksi target. Hal ini membantu kita memahami fitur mana yang memiliki pengaruh terbesar terhadap prediksi model.

- Visualisasi Feature Importance Logistic Regression



Feature importance untuk model Logistic Regression menunjukkan koefisien yang menunjukkan besarnya pengaruh setiap fitur terhadap probabilitas target. Fitur dengan koefisien positif meningkatkan probabilitas target, sedangkan fitur dengan koefisien negatif menurunkan probabilitas target.

- Visualisasi Feature Importance Decision Tree

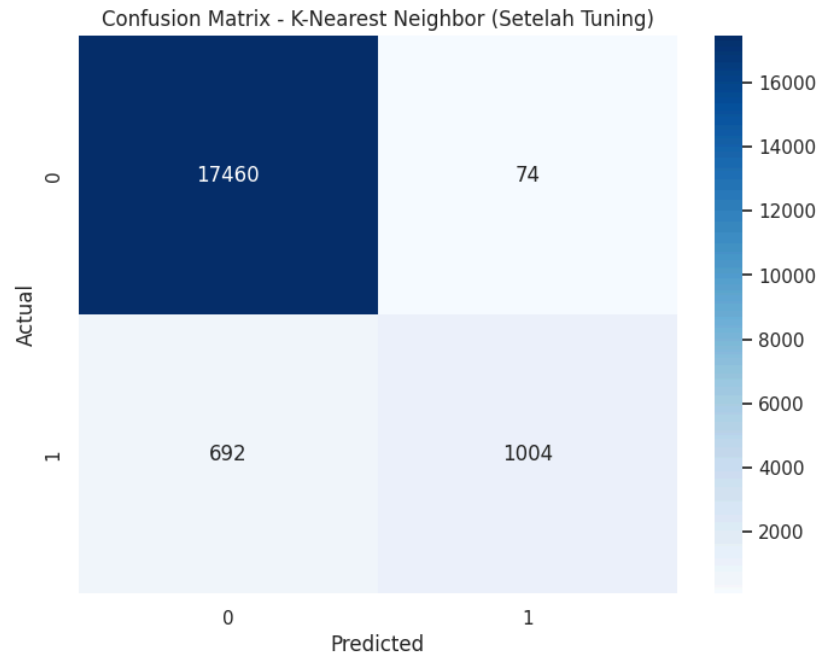


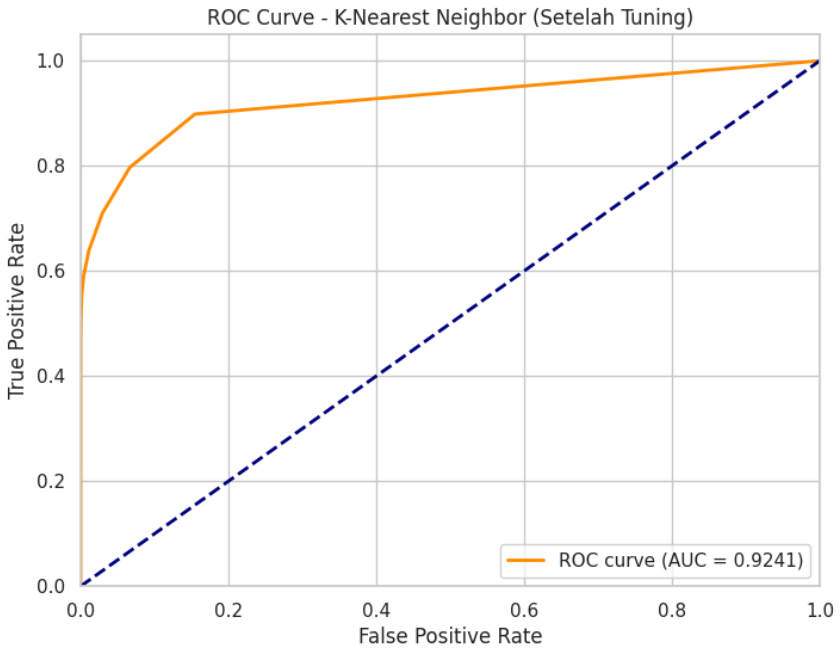
Feature importance untuk model Decision Tree menunjukkan seberapa banyak setiap fitur berkontribusi dalam mengurangi impurity dalam tree. Fitur dengan importance yang lebih tinggi lebih berkontribusi dalam membuat prediksi yang akurat.

3. Tuning Hyperparameter untuk Model Terbaik

Tuning hyperparameter adalah proses menemukan kombinasi hyperparameter yang menghasilkan performa model terbaik. Pada bagian ini, kita melakukan grid search untuk menemukan hyperparameter optimal untuk model terbaik berdasarkan akurasi.

```
Model terbaik berdasarkan akurasi: K-Nearest Neighbor dengan akurasi 0.9594
Hyperparameter terbaik: {'metric': 'manhattan', 'n_neighbors': 9, 'weights': 'uniform'}
Skor terbaik: 0.9598

Evaluasi Model Terbaik Setelah Tuning:
Accuracy: 0.9602
Precision: 0.9314
Recall: 0.5920
F1 Score: 0.7239
```





Tuning hyperparameter berhasil meningkatkan akurasi model K-Nearest Neighbor dari 0.9594 menjadi 0.9602. Model dengan hyperparameter optimal memiliki precision yang lebih tinggi tetapi recall yang sedikit lebih rendah dibandingkan model sebelumnya.

H. Kesimpulan

Berdasarkan analisis dataset Diabetes Prediction menggunakan algoritma supervised learning, dapat disimpulkan:

- Model terbaik adalah K-Nearest Neighbor dengan akurasi sebesar 96.02% setelah dilakukan tuning hyperparameter (metric: 'manhattan', n_neighbors: 9, weights: 'uniform').
- Fitur yang paling berpengaruh dalam prediksi diabetes adalah kadar HbA1c, kadar glukosa darah, dan usia, yang menunjukkan pentingnya parameter-parameter ini dalam skrining diabetes.
- Meskipun memiliki akurasi yang tinggi, model K-Nearest Neighbor masih menunjukkan trade-off antara precision (93.14%) dan recall (59.20%), yang menunjukkan bahwa model lebih baik dalam meminimalkan false positive daripada mendeteksi semua kasus positif.
- Penggunaan teknik preprocessing data seperti penanganan outlier, encoding fitur kategorikal, dan normalisasi data berperan penting dalam meningkatkan performa model.

Daftar Pustaka

Alzubaidi, A. A., Halawani, S. M., & Jarrah, M. (2023). Towards stacking ensemble for Diabetes Mellitus using combination of machine learning techniques. *IJACSA International Journal of Advanced Computer Science and Applications*, 14(12). https://pdfs.semanticscholar.org/a240/8111f40b81091dcda04d2068c054b89d1129.pdf

Alzubaidi, A. A., Halawani, S. M., & Jarrah, M. (2024). Integrated ensemble model for Diabetes Mellitus detection. *IJACSA International Journal of Advanced Computer Science and Applications*, 15(4). https://www.researchgate.net/publication/380387722_Integrated_Ensemble_Model_for_Diabetes_Mellitus_Detection

Kaliappan, J., Saravana Kumar, J. J., Sundaravelan, S., Anesh, T., Rithik, R. R., Singh, Y., Vera-Garcia, D. V., Himeur, Y., Mansoor, W., Atalla, S., & Srinivasan, K. (2024). Analyzing classification and feature selection strategies for diabetes prediction across diverse diabetes datasets. *Frontiers in Artificial Intelligence*, 7, 1421751. https://pmc.ncbi.nlm.nih.gov/articles/PMC11371799/