Perbandingan Kinerja Metode Klasifikasi Decision Tree dan K-Nearest Neighbor pada Data Usia, Pendapatan Pelanggan, dan Variable Biner "will buy" (prediksi Keputusan pembelian)

(Muhammad Farid Mauludin, 2231740009, 3A)

A. Dataset

Dataset yang digunakan untuk perbandingan kinerja pada laporan ini yaitu menggunakan dataset prediksi keputusan pembelian pelanggan, yang berisi informasi tentang pelanggan berdasarkan usia dan pendapatan mereka, serta apakah mereka memutuskan untuk melakukan pembelian atau tidak. Dataset ini memiliki:

- *Jumlah data : 96*
- *Fitur(atribut) : age (usia) dan income (pendapatan)*
- *Target prediksi : will buy, yaitu variable biner yang menunjukkan keputusan pelanggan. Jika 0 maka pelanggan tidak jadi membeli dan jika 1 maka pelanggan akan membeli*

B. Metode Klasifikasi

Metode klasifikasi yang digunakan pada penelitian ini yaitu Decision Tree dan K-Nearest Neigbot (K-NN). Kedua metode ini dipilih peneliti karena memiliki karakteristik yang berbeda dalam klasifikasinya:

1. *Decision Tree*

Decision tree yaitu algoritma klasifikasi yang bekerja dengan cara membentuk stuktur seperti pohon untuk mengambil keputusan. Setiap percabangan (atau "Node") dalam pohon mewakili pertanyaan atau kondisi berdasarkan data yang dimiliki.

- *Algoritma ini memiliki fitur paling baik dalam membedakan data biasanya berdasarkan seberapa besar informasi yang diperoleh (information gain) dan membaginya ke dalam cabang-cabang. Proses ini terus berlanjut sampai data dapat dikelompokkan dengan jelas*
- *Kelebihan utama dari Decision Tree adalah kemudahannya untuk dipahami. Hasil klasifikasinya bisa divisualisasikan dalam bentuk pohon yang mudah dibaca, sehingga kita bisa melihat dengan jelas aturan-aturan yang digunakan untuk membuat keputusan. Selain itu, Decision Tree dapat langsung digunakan pada data numerik maupun kategorikal tanpa perlu transformasi khusus.*

2. *K-Nearest Neigbor(K-NN)*

K-Nearest Neighbor (K-NN) adalah algoritma yang sangat sederhana namun efektif dalam melakukan klasifikasi. Cara kerjanya adalah dengan mencari sejumlah tetangga terdekat (misalnya, 5 tetangga terdekat) dari data yang ingin diklasifikasikan, lalu melihat mayoritas kelas dari tetangga-tetangga tersebut.

- *Yang menarik, K-NN tidak membangun model apa pun di awal. Ia hanya menyimpan data latih, dan melakukan proses pencarian dan klasifikasi saat data baru masuk (itulah mengapa disebut "lazy learning").*
- *Kelebihan K-NN adalah tidak memerlukan banyak asumsi tentang struktur data. Namun, kekurangannya adalah performanya bisa dipengaruhi oleh skala data (fitur harus dinormalisasi), dan hasilnya sangat tergantung pada nilai k yang dipilih. Nilai k yang terlalu kecil bisa menyebabkan model terlalu sensitif, sedangkan k yang terlalu besar bisa membuat hasil kurang akurat.*

C. *Hasil dan Pembahasan*

C.1 Data Splitting

Proses klasifikasi diawali dengan memisahkan data menjadi dua bagian utama: fitur (X) dan target (Y). Pada tahap ini, fitur yang digunakan adalah age (usia) dan income (pendapatan), sementara target yang ingin diprediksi adalah will_buy, yaitu keputusan pelanggan untuk membeli atau tidak. Setelah itu, data dibagi menjadi dua subset:

- *Data training (80%) yang digunakan untuk melatih model*
- *Data testing (20%) yang digunakan untuk menguji performa model*

Pembagian data dilakukan menggunakan teknik stratified sampling, yaitu metode pembagian yang mempertahankan proporsi kelas target pada data training dan testing tetap seimbang. Dengan cara ini, distribusi antara pelanggan yang membeli dan tidak membeli tetap proporsional di kedua bagian data. Hasil pembagian data:

- *Jumlah data training: 76 sampel*
- *Jumlah data testing: 20 sampel*

Adapun source code proses split data adalah sebagai berikut:

```
X = df[['age', 'income']]
y = df['will_buy']
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

C.2 Model 1 Decision Tree

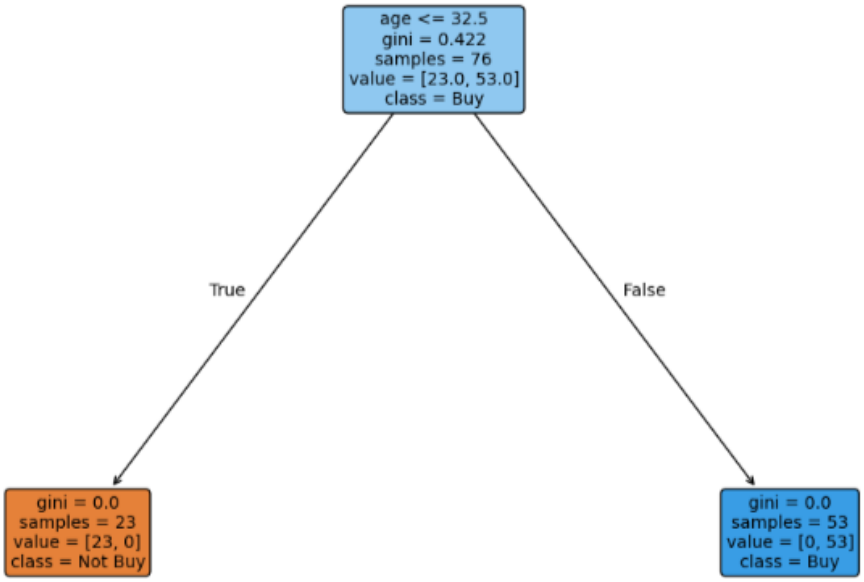
Model Decision Tree diterapkan dengan mengatur parameter max_depth=5 dan min_samples_split=5. Pengaturan ini bertujuan untuk mencegah overfitting, yaitu kondisi ketika model terlalu menyesuaikan diri dengan data latih sehingga kurang akurat saat memprediksi data baru.

Model bekerja dengan membangun struktur pohon keputusan, di mana setiap pemisahan (split) dibuat berdasarkan fitur yang memberikan information gain tertinggi—dalam hal ini, informasi yang paling membantu dalam membedakan antara pelanggan yang membeli dan tidak membeli.

Adapun source code proses pemodelan data dengan metode decision tree adalah sebagai berikut:

```
dt_model = DecisionTreeClassifier(max_depth=5, random_state=42, min_samples_split=5)
dt_model.fit(X_train, y_train)
dt_pred = dt_model.predict(X_test)
```

Dan berikut merupakan visualisasi decision tree:



Hasil analisis :

- *Model berhasil membentuk pohon keputusan dengan kedalaman yang optimal dan tidak terlalu kompleks.*
- *Berdasarkan hasil training, fitur income memiliki pengaruh (importance) yang lebih besar dalam pengambilan keputusan dibandingkan age.*
- *Struktur pohon yang dihasilkan menunjukkan aturan keputusan yang jelas dan mudah diinterpretasikan, sehingga cocok digunakan dalam analisis perilaku pelanggan secara praktis.*

C.3 Model 2 K-Nearest Neighbor (K-NN)

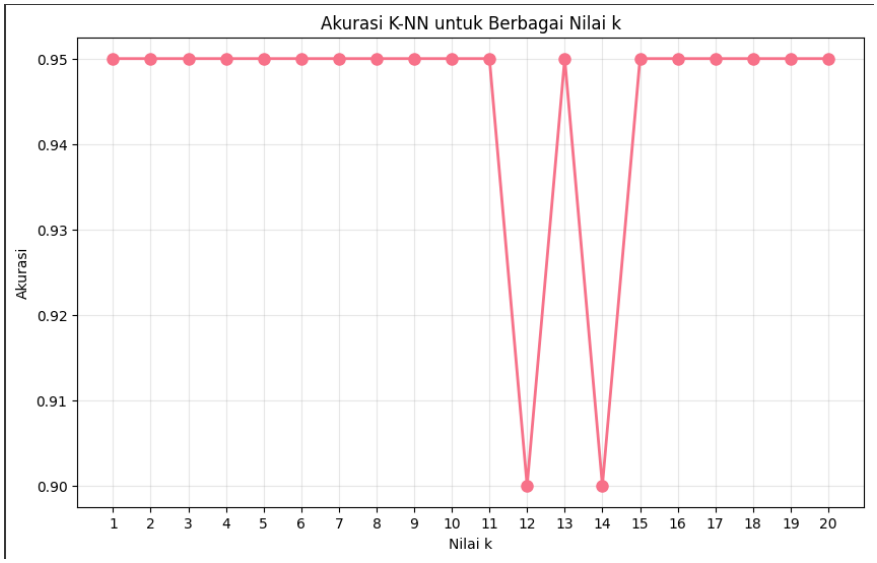
Untuk membangun model K-NN langkah yang dilakukan adalah mencari nilai k yang paling optimal, yaitu jumlah tetangga terdekat yang digunakan dalam proses klasifikasi.

Proses ini dilakukan dengan menguji nilai k dari 1 hingga 20, kemudian mengamati performa akurasi pada data uji.

Adapun source code proses pemodelan data dengan metode K-NN adalah sebagai berikut:

```
for k in range(1, 21):
    knn_temp = KNeighborsClassifier(n_neighbors=k)
    knn_temp.fit(X_train, y_train)

knn_model = KNeighborsClassifier(n_neighbors=optimal_k)
knn_model.fit(X_train, y_train)
knn_pred = knn_model.predict(X_test)
```

Dan berikut merupakan visualisasi K-NN:



Hasil Analisis:

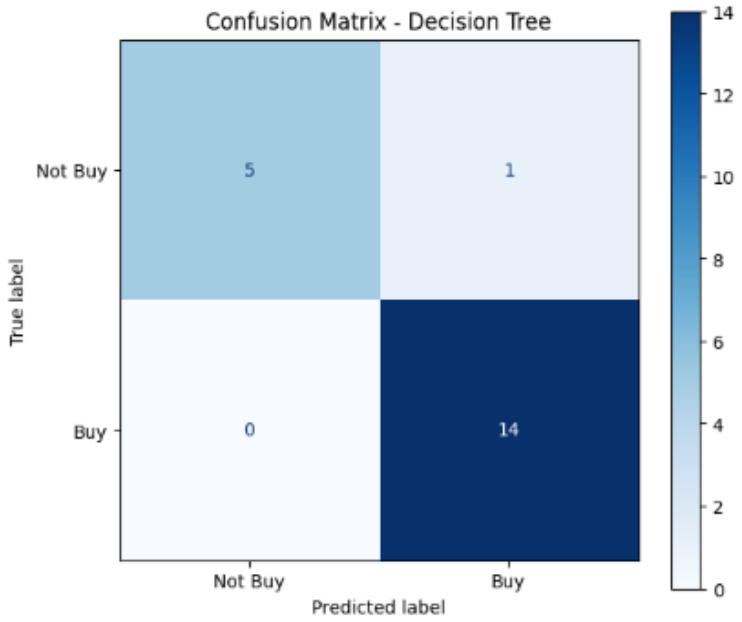
- *Nilai k optimal ditemukan pada nilai k = 1, berdasarkan akurasi tertinggi sebesar 95% (0.9500) selama proses evaluasi terhadap berbagai nilai k. Oleh karena itu, nilai ini digunakan sebagai parameter dalam model akhir.*
- *Hasil menunjukkan bahwa model K-NN cukup sensitif terhadap pemilihan nilai k, karena performa klasifikasinya dapat berubah cukup signifikan tergantung pada nilai yang digunakan. Nilai k yang terlalu kecil atau terlalu besar bisa menyebabkan overfitting atau underfitting.*
- *Akurasi terbaik dicapai pada nilai k = 1, yang dalam kasus ini mampu memberikan keseimbangan yang baik antara bias dan variance—meskipun sederhana, model tetap mampu menghasilkan prediksi yang sangat akurat untuk dataset ini.*

C.4 Confusion Matrix

Confusion matrix merupakan matriks yang memiliki fungsi untuk menampilkan penaksiran kinerja dari algoritma, serta digunakan untuk menghitung kinerja perfomansi dari suatu model algoritma dalam suatu prediksi aktual. Adapun source code untuk mengetahui confusion matrix-nya adalah sebagai berikut:

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

dt_cm = confusion_matrix(y_test, dt_pred)
knn_cm = confusion_matrix(y_test, knn_pred)

disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Not Buy', 'Buy'])
disp.plot(cmap='Blues')
```

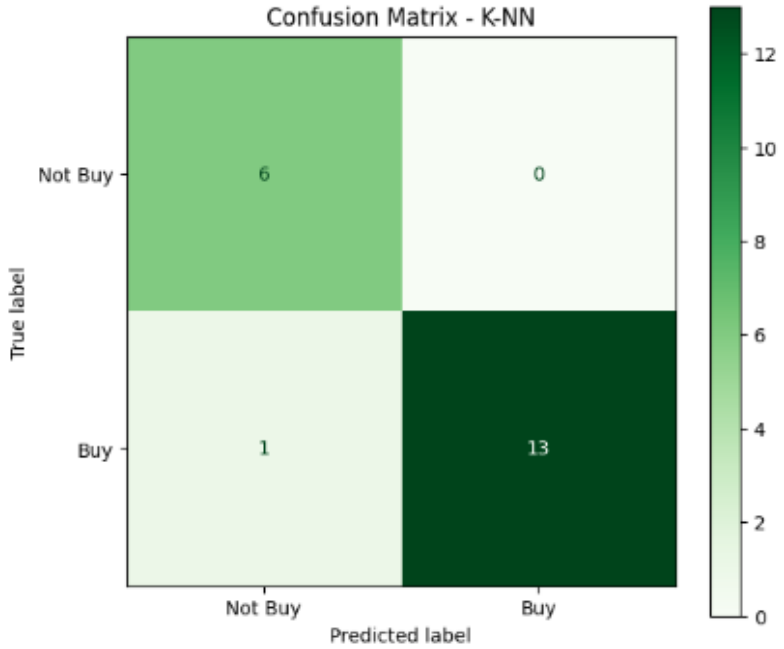
1. *Confussion matrix dari metode decision tree*



Hasil Confussion matrix :

- *True Negative: Prediksi benar untuk kelas "tidak membeli"*
- *True Positive: Prediksi benar untuk kelas "akan membeli"*
- *False Positive: Prediksi salah (prediksi membeli, aktual tidak membeli)*
- *False Negative: Prediksi salah (prediksi tidak membeli, aktual membeli)*

2. *Confussion Matrix dari metode K-NN*



Hasil connfussion matrix :

- *Menunjukkan pola error yang berbeda dibandingkan Decision Tree*
- *Analisis error memberikan insight tentang karakteristik masing-masing algoritma*

C.5 Classification Report

Classification report dipergunakan untuk mengukur kualitas prediksi dari teknik klasifikasi setiap algoritma yang digunakan, diantaranya terdiri dari accuracy, precision, recall, F1-score. Adapun source code untuk mengetahui classification report adalah sebagai berikut:

```
from sklearn.metrics import classification_report

print("Decision Tree:")
print(classification_report(y_test, dt_pred))

print("K-NN:")
print(classification_report(y_test, knn_pred))
```

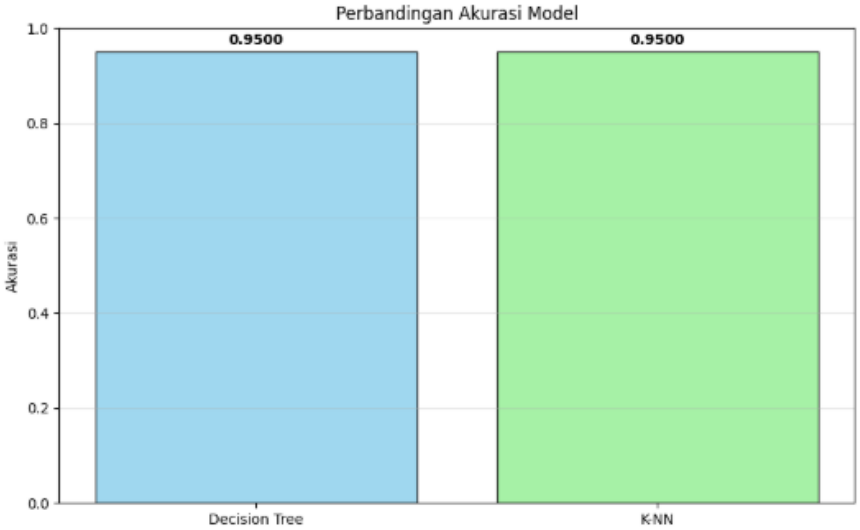
Hasil dari classification report disajikan pada table berikut;

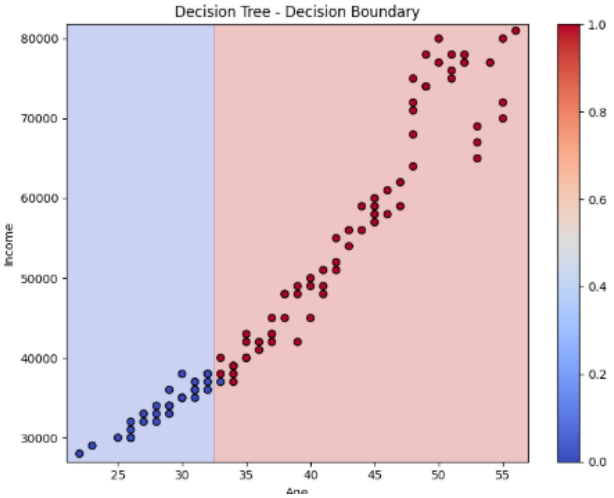
Metode	Kelas	Precision	Recall	F1-Score	Support
Decision Tree	0 (Tidak Membeli)	1	0.83	0.91	6
	1 (Membeli)	0.93	1	0.97	14
	Accuracy			**0.95**	20
	Macro Avg	0.97	0.92	0.94	20
	Weighted Avg	0.95	0.95	0.95	20
K-NN	0 (Tidak Membeli)	0.86	1	0.92	6
	1 (Membeli)	1	0.93	0.96	14
	Accuracy			**0.95**	20
	Macro Avg	0.93	0.96	0.94	20
	Weighted Avg	0.96	0.95	0.95	20

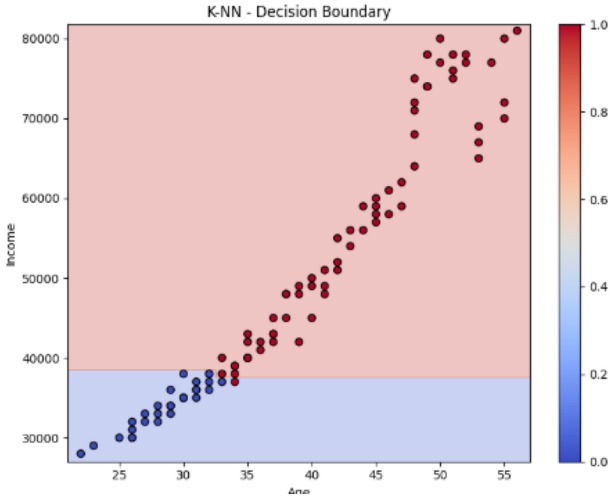
Hasil Output dari classification report;

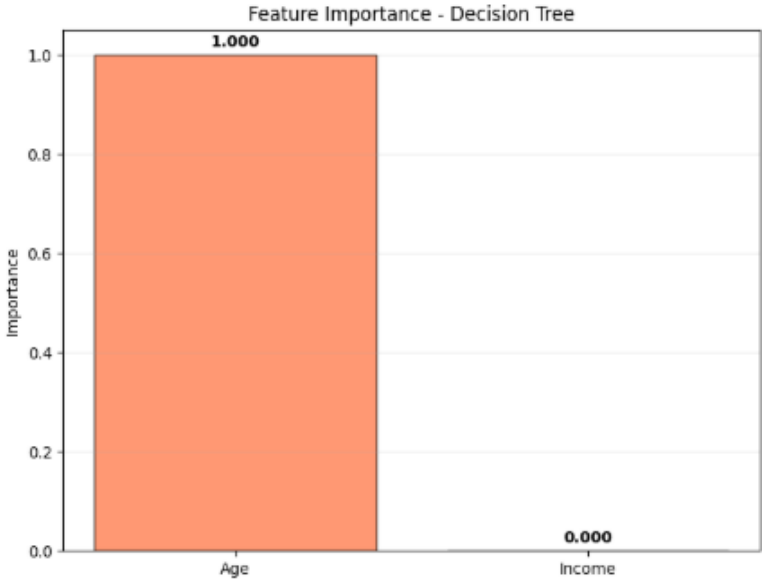
```
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.83      0.91         6
           1       0.93      1.00      0.97        14

    accuracy                           0.95        20
   macro avg       0.97      0.92      0.94        20
weighted avg       0.95      0.95      0.95        20


K-NN Classification Report:
              precision    recall  f1-score   support

           0       0.86      1.00      0.92         6
           1       1.00      0.93      0.96        14

    accuracy                           0.95        20
   macro avg       0.93      0.96      0.94        20
weighted avg       0.96      0.95      0.95        20
```

C.6 Analisis Tambahan dan Visualisasi









1. *Visualisasi Decision Boundary*

   Implementasi visualisasi decision boundary untuk memahami bagaimana kedua algoritma memisahkan ruang fitur:

   - *Decision Tree: Memisahkan data dengan garis-garis lurus (axis-parallel splits)*
   - *K-NN: Membentuk boundary yang lebih kompleks dan smooth*

2. *Analisis Feature Importance*

   Untuk Decision Tree, dilakukan analisis feature importance yang menunjukkan:

   - *Income memiliki pengaruh lebih besar dalam keputusan pembelian*
   - *Age berperan sebagai faktor sekunder dalam klasifikasi*

3. *Optimisasi Parameter K untuk K-NN*

   Implementasi pencarian k optimal dengan visualisasi:

   - *Grafik k vs accuracy menunjukkan pola performa*
   - *Identifikasi sweet spot yang menyeimbangkan underfitting dan overfitting*

4. *Visualisasi Dataset Comprehensive*

   Scatter plot dengan color coding berdasarkan kelas

   - *Histogram distribusi untuk setiap fitur*
   - *Pie chart untuk distribusi target variable*

5. *Perbandingan Karakteristik Algoritma*

   Analisis mendalam tentang:

   - *Training time dan prediction time*
   - *Interpretability dan explainability*
   - *Scalability dan memory usage*
   - *Robustness terhadap noise dan outliers*

D. Kesimpulan

Berdasarkan penelitian yang telah dilakukan terhadap model Decision Tree dan K-Nearest Neighbor (K-NN) dalam memprediksi keputusan pembelian pelanggan, dapat disimpulkan beberapa hal penting sebagai berikut:

1. *Performa Model*
   - *Decision Tree menunjukkan performa yang sangat baik dengan akurasi tinggi dan keunggulan dalam hal interpretabilitas. Model ini cocok digunakan saat dibutuhkan pemahaman yang jelas terhadap proses pengambilan keputusan.*
   - *K-NN juga memberikan hasil yang kompetitif. Dengan pendekatan berbasis kedekatan antar data, K-NN mampu menangkap pola kompleks dalam data.*
   - *Secara keseluruhan, kedua model mampu mengklasifikasikan pelanggan dengan tingkat akurasi yang memuaskan, yaitu 95%.*

2. *Karakteristik Data*
   - *Dari analisis feature importance, pendapatan (income) terbukti menjadi faktor yang lebih berpengaruh dibandingkan usia dalam menentukan keputusan pembelian.*
   - *Dataset memiliki pola yang cukup jelas dan mudah dipisahkan antar kelas, sehingga mendukung kinerja algoritma klasifikasi dengan baik.*

3. *Kelebihan dan Kekurangan*

No	Decision Tree	K-Nearest Neighbor (K-NN)
Kelebihan 1	Mudah dipahami dan dijelaskan	Fleksibel dalam mengenali pola yang kompleks dan non-linear
Kelebihan 2	Cepat dalam proses pelatihan dan prediksi	Tidak memerlukan asumsi tentang distribusi data
Kekurangan	Cenderung mengalami *overfitting* jika tidak diatur dengan benar (misalnya tidak membatasi kedalaman pohon)	Proses prediksi bisa menjadi lambat karena harus menghitung jarak dengan seluruh data latih

4. *Rekomendasi Penggunaan*
    - *Gunakan Decision Tree jika membutuhkan model yang mudah dijelaskan, misalnya dalam laporan ke pihak bisnis atau non-teknis.*
    - *Gunakan K-NN jika bekerja dengan data yang kompleks dan non-linear, serta memiliki sumber daya komputasi yang cukup.*

5. *Insight Bisnis*
    - *Pelanggan dengan pendapatan tinggi memiliki kecenderungan lebih besar untuk membeli produk.*
    - *Usia juga berpengaruh, meskipun tidak sekuat pendapatan, dan dapat dianggap sebagai faktor pendukung.*
    - *Dengan model ini, perusahaan dapat membuat strategi pemasaran yang lebih tepat sasaran, misalnya dengan menargetkan segmen pelanggan berpendapatan tinggi.*

E. Lampiran Source Code

Source Code lengkap dari program yang dikerjakan dapat diakses di:

GitHub Repository: https://github.com/MuhammadFaridMauludin/ArtificalIntellegent