LAPORAN CAPSTONE PROJECT



MANAJEMEN BIG DATA

Pengampu Dr. Eng. Farrikh Alzami, M.Kom

Program Sarjana Sistem Informasi

Disusun oleh

|  |  |
| --- | --- |
| NIM | NAMA |
|  |  |
|  |  |
|  |  |

UNIVERSITAS DIAN NUSWANTORO

SEMARANG, 2025

DAFTAR ISI

[LAPORAN CAPSTONE PROJECT PREDIKSI TINGKAT PENDAPATAN MENGGUNAKAN ALGORITMA DECISION TREE Studi Kasus Dataset Adult Census Income 1](#_Toc199267083)

[ABSTRAK 1](#_Toc199267084)

[1. PENDAHULUAN 1](#_Toc199267085)

[1.1 Latar Belakang 1](#_Toc199267086)

[1.2 Rumusan Masalah 2](#_Toc199267087)

[1.3 Tujuan Penelitian 2](#_Toc199267088)

[1.4 Manfaat Penelitian 2](#_Toc199267089)

[1.5 Batasan Penelitian 2](#_Toc199267090)

[2. TINJAUAN PUSTAKA 2](#_Toc199267091)

[2.1 Machine Learning untuk Prediksi Pendapatan 2](#_Toc199267092)

[2.2 Algoritma Decision Tree 2](#_Toc199267093)

[2.3 Dataset Adult Census Income 3](#_Toc199267094)

[3. METODOLOGI PENELITIAN 3](#_Toc199267095)

[3.1 Kerangka Kerja Penelitian 3](#_Toc199267096)

[3.2 Dataset dan Fitur 3](#_Toc199267097)

[3.3 Tahapan Preprocessing 4](#_Toc199267098)

[3.4 Model Development 5](#_Toc199267099)

[3.5 Evaluasi Model 5](#_Toc199267100)

[4. IMPLEMENTASI 5](#_Toc199267101)

[4.1 Environment dan Tools 5](#_Toc199267102)

[4.2 Pipeline Implementation 6](#_Toc199267103)

[4.3 Web Application Development 8](#_Toc199267104)

[5. HASIL DAN PEMBAHASAN 9](#_Toc199267105)

[5.1 Hasil Eksplorasi Data 9](#_Toc199267106)

[5.2 Hasil Preprocessing 9](#_Toc199267107)

[5.3 Hasil Model Training 10](#_Toc199267108)

[5.4 Analisis Model 11](#_Toc199267109)

[5.5 Hasil Implementasi Web Application 12](#_Toc199267110)

[6. KESIMPULAN DAN SARAN 13](#_Toc199267111)

[6.1 Kesimpulan 13](#_Toc199267112)

[6.2 Kontribusi Penelitian 13](#_Toc199267113)

[6.3 Limitasi Penelitian 14](#_Toc199267114)

[6.4 Saran untuk Penelitian Selanjutnya 14](#_Toc199267115)

[DAFTAR PUSTAKA 15](#_Toc199267116)

[LAMPIRAN 15](#_Toc199267117)

[Lampiran A: Source Code 15](#_Toc199267118)

[Lampiran B: Dataset Description 15](#_Toc199267119)

[Lampiran C: Additional Visualizations 16](#_Toc199267120)

[Lampiran D: Model Artifacts 16](#_Toc199267121)

# LAPORAN CAPSTONE PROJECT PREDIKSI TINGKAT PENDAPATAN MENGGUNAKAN ALGORITMA DECISION TREE Studi Kasus Dataset Adult Census Income

## ABSTRAK

**Latar Belakang:** Prediksi tingkat pendapatan merupakan salah satu aplikasi penting dalam bidang data mining dan machine learning, khususnya untuk analisis ekonomi, kebijakan sosial, dan pengambilan keputusan bisnis.

**Tujuan:** Mengembangkan model klasifikasi untuk memprediksi apakah seseorang memiliki pendapatan di atas atau di bawah $50,000 per tahun berdasarkan karakteristik demografis dan pekerjaan.

**Metode:** Penelitian ini menggunakan algoritma Decision Tree dengan dataset Adult Census Income dari UCI Machine Learning Repository. Proses meliputi eksplorasi data, preprocessing, feature engineering, hyperparameter tuning, dan evaluasi model.

**Hasil:** Model Decision Tree dengan hyperparameter optimal mencapai akurasi **86.08%** pada data testing dengan feature importance tertinggi pada **fitur-fitur kunci seperti status perkawinan, perolehan modal, dan tingkat pendidikan**.

**Kesimpulan:** Model yang dikembangkan dapat digunakan sebagai alat bantu prediksi pendapatan dengan tingkat akurasi yang baik dan telah diimplementasikan dalam aplikasi web menggunakan Streamlit.

**Kata Kunci:** Machine Learning, Decision Tree, Classification, Income Prediction, Data Mining

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Prediksi tingkat pendapatan seseorang berdasarkan karakteristik demografis dan pekerjaan memiliki berbagai aplikasi praktis dalam dunia nyata. Informasi ini dapat digunakan oleh:

* **Pemerintah**: Untuk perencanaan kebijakan sosial dan ekonomi
* **Institusi Keuangan**: Untuk penilaian kredit dan produk finansial
* **Perusahaan**: Untuk segmentasi pasar dan strategi marketing
* **Peneliti Sosial**: Untuk memahami faktor-faktor yang mempengaruhi tingkat pendapatan

### 1.2 Rumusan Masalah

1. Bagaimana mengembangkan model klasifikasi yang akurat untuk memprediksi tingkat pendapatan seseorang?
2. Fitur apa saja yang paling berpengaruh terhadap prediksi tingkat pendapatan?
3. Bagaimana implementasi model dalam aplikasi yang user-friendly?

### 1.3 Tujuan Penelitian

#### Tujuan Umum:

Mengembangkan sistem prediksi tingkat pendapatan menggunakan algoritma machine learning.

#### Tujuan Khusus:

1. Melakukan eksplorasi dan analisis dataset Adult Census Income
2. Mengimplementasikan preprocessing dan feature engineering yang optimal
3. Membangun model Decision Tree dengan hyperparameter tuning
4. Mengevaluasi performa model menggunakan berbagai metrik
5. Mengimplementasikan model dalam aplikasi web menggunakan Streamlit

### 1.4 Manfaat Penelitian

* **Akademis**: Kontribusi dalam bidang applied machine learning untuk prediksi ekonomi
* **Praktis**: Menyediakan alat bantu prediksi yang dapat digunakan berbagai stakeholder
* **Teknologi**: Demonstrasi implementasi end-to-end machine learning pipeline

### 1.5 Batasan Penelitian

* Dataset terbatas pada data census tahun 1994
* Fokus pada algoritma Decision Tree
* Evaluasi dilakukan pada single algorithm (tidak comparative study)

## 2. TINJAUAN PUSTAKA

### 2.1 Machine Learning untuk Prediksi Pendapatan

[Isi dengan referensi penelitian terkait prediksi income menggunakan ML]

### 2.2 Algoritma Decision Tree

Decision Tree merupakan algoritma supervised learning yang dapat digunakan untuk masalah klasifikasi dan regresi. Keunggulan algoritma ini:

* **Interpretabilitas tinggi**: Hasil mudah dipahami dan dijelaskan
* **Tidak memerlukan asumsi distribusi**: Non-parametric algorithm
* **Dapat menangani data numerik dan kategorikal**
* **Otomatis feature selection**: Mengidentifikasi fitur yang paling penting

#### 2.2.1 Cara Kerja Decision Tree

1. **Splitting**: Memilih fitur dan threshold terbaik untuk membagi data
2. **Criterion**: Menggunakan Gini impurity atau Entropy untuk mengukur kualitas split
3. **Pruning**: Mencegah overfitting dengan membatasi kedalaman atau jumlah sampel minimum

### 2.3 Dataset Adult Census Income

Dataset ini berasal dari UCI Machine Learning Repository dan berisi data census tahun 1994. Dataset terdiri dari:

* **32,561 instances** dengan **14 atribut** (sebelum penambahan kolom nama dan pemrosesan)
* **Target variable**: Income (≤50K atau >50K)
* **Missing values**: Ada pada beberapa atribut kategorikal (workclass, occupation, native\_country).

## 3. METODOLOGI PENELITIAN

### 3.1 Kerangka Kerja Penelitian

Data Collection → EDA → Preprocessing → Feature Engineering →   
Model Training → Hyperparameter Tuning → Evaluation → Deployment

### 3.2 Dataset dan Fitur

#### 3.2.1 Deskripsi Dataset

* **Sumber**: UCI Machine Learning Repository
* **URL**: https://archive.ics.uci.edu/dataset/2/adult
* **Ukuran**: 32,561 records, 14 features + 1 target (sesuai deskripsi awal, notebook menambahkan nama kolom sehingga menjadi 15)
* **Type**: Supervised Classification

#### 3.2.2 Deskripsi Fitur

| Fitur | Type | Deskripsi |
| --- | --- | --- |
| age | Numeric | Usia individu |
| workclass | Categorical | Jenis pekerjaan |
| fnlwgt | Numeric | Final weight (representasi populasi) |
| education | Categorical | Level pendidikan |
| education\_num | Numeric | Pendidikan dalam angka |
| marital\_status | Categorical | Status pernikahan |
| occupation | Categorical | Jenis profesi |
| relationship | Categorical | Status hubungan keluarga |
| race | Categorical | Ras |
| sex | Categorical | Jenis kelamin |
| capital\_gain | Numeric | Keuntungan modal |
| capital\_loss | Numeric | Kerugian modal |
| hours\_per\_week | Numeric | Jam kerja per minggu |
| native\_country | Categorical | Negara asal |
| income | Categorical | Target: ≤50K atau >50K |

### 3.3 Tahapan Preprocessing

#### 3.3.1 Eksplorasi Data Awal (EDA)

1. **Dataset Overview**: Shape, info, dan statistik deskriptif
2. **Missing Values Analysis**: Identifikasi dan handling missing data
3. **Distribution Analysis**: Visualisasi distribusi setiap fitur
4. **Correlation Analysis**: Analisis korelasi antar fitur numerik
5. **Target Distribution**: Analisis balance/imbalance target class

#### 3.3.2 Data Cleaning

1. **Missing Values Handling**:
   * Categorical: Mode imputation
   * Numerical: Median imputation (meskipun pada dataset ini semua fitur numerik tidak memiliki missing values awal)
2. **Duplicate Removal**: Menghapus record yang duplikat
3. **Outlier Detection**: Identifikasi outlier pada fitur numerik (langkah ini tidak secara eksplisit dilakukan dengan penghapusan di notebook, namun distribusi diamati).

#### 3.3.3 Feature Engineering

1. **Label Encoding**: Konversi fitur kategorikal menjadi numerik menggunakan mapping manual.
2. **Feature Selection**: Penghapusan fitur education (karena education\_num lebih informatif dan numerik). Analisis korelasi tinggi dilakukan, namun tidak ada fitur yang dihapus berdasarkan threshold > 0.8.
3. **Feature Importance Analysis**: Analisis kontribusi setiap fitur setelah model dilatih.

### 3.4 Model Development

#### 3.4.1 Data Splitting

* **Training Set**: 90% dari total data (29,283 sampel)
* **Testing Set**: 10% dari total data (3,254 sampel)
* **Stratified Sampling**: Mempertahankan proporsi target class pada saat splitting.

#### 3.4.2 Hyperparameter Tuning

Menggunakan GridSearchCV dengan parameter:

param\_grid = {  
 'max\_depth': [None, 5, 10, 15, 20],  
 'min\_samples\_split':,  
 'min\_samples\_leaf':,  
 'criterion': ['gini', 'entropy']  
}

#### 3.4.3 Cross Validation

* **K-Fold CV**: 3-fold cross validation digunakan dalam GridSearchCV.
* **Scoring Metric**: Accuracy
* **Parallel Processing**: n\_jobs=-1 untuk optimasi waktu training.

### 3.5 Evaluasi Model

#### 3.5.1 Metrik Evaluasi

1. **Accuracy**: Overall correctness
2. **Precision**: True Positive Rate per class (dan agregat)
3. **Recall**: Sensitivity per class (dan agregat)
4. **F1-Score**: Harmonic mean of precision and recall (per class dan agregat)
5. **Confusion Matrix**: Detailed classification results

#### 3.5.2 Analisis Hasil

1. **Feature Importance**: Ranking fitur berdasarkan kontribusi
2. **Model Interpretability**: Visualisasi decision tree
3. **Performance Analysis**: Training vs Testing performance

## 4. IMPLEMENTASI

### 4.1 Environment dan Tools

* **Python Version**: 3.10.14 (berdasarkan metadata notebook, dapat disesuaikan jika berbeda)
* **Key Libraries**:
  + pandas, numpy: Data manipulation
  + scikit-learn: Machine learning
  + matplotlib, seaborn: Visualization
  + streamlit: Web deployment
  + plotly: Interactive visualization (digunakan di aplikasi Streamlit)
  + joblib: Penyimpanan model

### 4.2 Pipeline Implementation

#### 4.2.1 Data Loading dan EDA

# Kode untuk loading data dan EDA  
import pandas as pd  
  
# Load dataset  
column\_names = ['age', 'workclass', 'fnlwgt', 'education', 'education\_num', 'marital\_status',   
 'occupation', 'relationship', 'race', 'sex', 'capital\_gain', 'capital\_loss',   
 'hours\_per\_week', 'native\_country', 'income']  
# Diasumsikan file 'adult.data' ada di direktori yang sama  
df = pd.read\_csv('adult.data', names=column\_names, skipinitialspace=True, na\_values="?")  
  
# Dataset Overview  
print("Dataset Overview:")  
print(f"Shape: {df.shape}")  
# print(df.info()) # Untuk ringkasan lebih detail  
  
# Check Missing Values  
missing\_values = df.isnull().sum()  
print("\nMissing Values:")  
print(missing\_values[missing\_values > 0])  
  
# Target Distribution  
# print("\nTarget Distribution:")  
# print(df['income'].value\_counts(normalize=True))

#### 4.2.2 Preprocessing Pipeline

# Kode untuk preprocessing  
  
# Handle missing values (contoh dari notebook)  
for column in ['workclass', 'occupation', 'native\_country']: # Kolom dengan missing values  
 if df[column].dtype == 'object':  
 df[column].fillna(df[column].mode(), inplace=True) # Menggunakan mode() untuk menghindari Series  
 # else: # Fitur numerik tidak punya missing values di awal  
 # df[column].fillna(df[column].median(), inplace=True)  
  
# Remove duplicates  
# initial\_rows = len(df)  
df.drop\_duplicates(inplace=True)  
# print(f"Removed {initial\_rows - len(df)} duplicate rows.")  
  
# Encoding for 'sex' (contoh)  
sex\_map = {'Male':1, 'Female':0}  
df['sex'] = df['sex'].map(sex\_map)  
  
# Drop education column (karena education\_num sudah ada)  
if 'education' in df.columns:  
 df = df.drop('education', axis=1)

#### 4.2.3 Model Training

# Kode untuk training model  
from sklearn.model\_selection import train\_test\_split, GridSearchCV  
from sklearn.tree import DecisionTreeClassifier  
  
# Asumsikan df sudah di-encode semua fitur kategorikalnya termasuk target 'income'  
# dan X, y, X\_train, X\_test, y\_train, y\_test sudah didefinisikan sebelumnya.  
# (Proses encoding lengkap dan split data ada di notebook bagian 4 dan 7)  
  
# Definisikan ulang X dan y untuk kejelasan dalam contoh ini  
# X = df.drop('income', axis=1)   
# y = df['income']  
# X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.1, random\_state=42, stratify=y)  
  
  
# Define parameter grid for GridSearchCV  
param\_grid = {  
 'max\_depth': [None, 5, 10, 15, 20],  
 'min\_samples\_split':,  
 'min\_samples\_leaf':,  
 'criterion': ['gini', 'entropy']  
}  
  
# Create decision tree classifier  
dt\_classifier = DecisionTreeClassifier(random\_state=42)  
  
# Perform grid search  
grid\_search = GridSearchCV(  
 estimator=dt\_classifier,  
 param\_grid=param\_grid,  
 cv=3, # Sesuai notebook  
 scoring='accuracy',  
 verbose=1, # Sesuai notebook  
 n\_jobs=-1   
)  
# grid\_search.fit(X\_train, y\_train) # Proses training  
# best\_model = grid\_search.best\_estimator\_  
# print(f"Best Parameters: {grid\_search.best\_params\_}")

*(Catatan: Untuk contoh kode training, beberapa bagian seperti encoding lengkap dan split data di-comment atau disederhanakan agar lebih ringkas sebagai contoh di laporan. Kode penuh ada di notebook.)*

### 4.3 Web Application Development

#### 4.3.1 Streamlit Architecture

* **Frontend**: Streamlit components untuk input dan visualization. Menggunakan st.form untuk input pengguna, st.columns untuk layout, st.number\_input, st.selectbox untuk field input. Visualisasi menggunakan plotly.graph\_objects (untuk Gauge chart) dan plotly.express (untuk Bar chart).
* **Backend**: Model inference menggunakan joblib untuk memuat model Decision Tree yang telah dilatih dan komponen preprocessing lainnya (seperti encoding maps dan feature names). Fungsi predict\_income menangani preprocessing data input dan prediksi.
* **Features**:
  + User-friendly input forms untuk 13 fitur.
  + Real-time prediction (setelah tombol “Predict” ditekan).
  + Interactive visualizations: Gauge chart untuk confidence level, bar chart untuk distribusi probabilitas kelas, dan bar chart untuk feature importance model.
  + Export functionality untuk mengunduh hasil prediksi terakhir dalam format JSON.
  + Tombol Reset untuk mengembalikan nilai input ke default.

#### 4.3.2 Deployment Features

1. **Input Validation**: Error handling untuk input user dilakukan melalui fungsi validate\_inputs yang memeriksa batasan nilai wajar untuk fitur numerik seperti age, education\_num, hours\_per\_week, capital\_gain, capital\_loss, dan fnlwgt. Pesan error ditampilkan jika validasi gagal.
2. **Model Loading**: Caching digunakan (@st.cache\_resource) untuk memuat model dan komponennya sekali saja, mengoptimalkan performance saat aplikasi diakses berulang kali.
3. **Visualization**: Interactive charts dari Plotly digunakan untuk menampilkan hasil prediksi (confidence gauge, probability distribution) dan feature importance model, memungkinkan pemahaman yang lebih baik bagi pengguna.
4. **Export Function**: Fungsi export\_prediction membuat file JSON yang berisi timestamp, data input, dan hasil prediksi (kelas, confidence, raw prediction). Pengguna dapat mengunduhnya melalui st.download\_button.

## 5. HASIL DAN PEMBAHASAN

### 5.1 Hasil Eksplorasi Data

#### 5.1.1 Dataset Overview

Dataset awal terdiri dari **32,561 baris** dan **15 kolom** (setelah penambahan nama kolom, termasuk kolom target income). Terdapat 6 fitur numerik (age, fnlwgt, education\_num, capital\_gain, capital\_loss, hours\_per\_week) dan 9 fitur objek/kategorikal (workclass, education, marital\_status, occupation, relationship, race, sex, native\_country, income). Statistik deskriptif menunjukkan rentang nilai, mean, median, dan standar deviasi untuk fitur numerik. Misalnya, usia rata-rata adalah sekitar 38.58 tahun. Untuk fitur kategorikal, frekuensi kelas tertinggi juga diamati, contohnya workclass terbanyak adalah ‘Private’ dan income mayoritas adalah ‘<=50K’.

#### 5.1.2 Missing Values Analysis

Terdapat missing values pada tiga fitur kategorikal:

* workclass: 1836 missing values (5.64% dari total data)
* occupation: 1843 missing values (5.66% dari total data)
* native\_country: 583 missing values (1.79% dari total data) Fitur-fitur numerik tidak memiliki missing values.

#### 5.1.3 Feature Distribution

Visualisasi distribusi fitur dilakukan menggunakan histogram untuk fitur numerik dan bar plot untuk fitur kategorikal (gambar numeric\_histograms.png dan distribution\_[feature].png dihasilkan oleh notebook). Observasi dari distribusi ini membantu memahami skewness, rentang, dan modus dari setiap fitur. Misalnya, capital\_gain dan capital\_loss sebagian besar bernilai nol. Distribusi kelas target income menunjukkan ketidakseimbangan, dengan sekitar 75.9% data memiliki pendapatan ‘<=50K’ dan 24.1% ‘>50K’.

#### 5.1.4 Correlation Analysis

Analisis korelasi awal antar fitur numerik dilakukan menggunakan heatmap (gambar correlation\_matrix.png). Ini membantu mengidentifikasi hubungan linear awal antar variabel numerik sebelum dilakukan encoding pada fitur kategorikal.

### 5.2 Hasil Preprocessing

#### 5.2.1 Data Cleaning Results

* **Original Shape**: (32561, 15)
* **After Cleaning**: (32537, 15) setelah penghapusan duplikat. Shape akhir setelah feature engineering dan sebelum splitting menjadi (32537, 14) yaitu 13 fitur input dan 1 target.
* **Removed Duplicates**: **24** records duplikat dihapus dari dataset.
* **Missing Values**: Missing values pada fitur workclass, occupation, dan native\_country ditangani menggunakan imputasi modus (nilai yang paling sering muncul) untuk masing-masing kolom.

#### 5.2.2 Feature Engineering Results

* **Encoding**: Seluruh fitur kategorikal (workclass, marital\_status, occupation, relationship, race, sex, native\_country) diubah menjadi representasi numerik menggunakan pemetaan manual (manual label encoding). Target variabel income juga diubah menjadi 0 (‘<=50K’) dan 1 (‘>50K’).
* **Feature Removal**: Fitur education dihapus karena informasinya sudah tercakup dalam education\_num yang bersifat numerik dan lebih sesuai untuk model. Tidak ada fitur lain yang dihapus karena korelasi tinggi antar fitur input (berdasarkan threshold 0.8 pada matriks korelasi setelah encoding, karena corr\_features hasilnya set()).
* **Final Feature Set**: Dataset final untuk pemodelan terdiri dari 13 fitur input: age, workclass, fnlwgt, education\_num, marital\_status, occupation, relationship, race, sex, capital\_gain, capital\_loss, hours\_per\_week, dan native\_country.

### 5.3 Hasil Model Training

#### 5.3.1 Hyperparameter Optimization

* **Best Parameters**: Hasil optimasi menggunakan GridSearchCV menunjukkan parameter terbaik untuk model Decision Tree adalah: {'criterion': 'gini', 'max\_depth': 10, 'min\_samples\_leaf': 4, 'min\_samples\_split': 10}.
* **Cross Validation Score**: Skor akurasi rata-rata dari 3-fold cross-validation pada data training dengan parameter terbaik adalah **[Tidak secara eksplisit tercetak di output, namun ini adalah nilai grid\_search.best\_score\_ yang mengarah pada pemilihan best\_params\_]**.
* **Training Time**: [Waktu training] (tidak tersedia dari output).

#### 5.3.2 Model Performance

Performa model dievaluasi pada data training dan data testing:

| Metric | Training Set | Testing Set |
| --- | --- | --- |
| Accuracy | 86.50% | 86.08% |
| Precision | 86.50% (micro avg) | 85.50% (weighted avg) |
| Recall | 86.50% (micro avg) | 86.08% (weighted avg) |
| F1-Score | 86.50% (micro avg) | 85.71% (weighted avg) |

*(Catatan: Nilai Precision dan F1-Score untuk Testing Set di atas dihitung dari Classification Report: Precision (weighted) = (0.89*2470 + 0.75*784)/(2470+784) = 0.855; F1-Score (weighted) = (0.91*2470 + 0.69*784)/(2470+784) = 0.8571. Recall (weighted) sesuai dengan akurasi karena supportnya sama.)*

Detail performa per kelas pada data testing:

* Kelas 0 (‘<=50K’): Precision 0.89, Recall 0.93, F1-Score 0.91
* Kelas 1 (‘>50K’): Precision 0.75, Recall 0.64, F1-Score 0.69

#### 5.3.3 Confusion Matrix

Confusion matrix pada data testing adalah sebagai berikut:

[[2301 169]  
 [ 284 500]]

Interpretasi:

* **True Negative (TN)**: 2301 individu dengan pendapatan <=50K diprediksi dengan benar.
* **False Positive (FP)**: 169 individu dengan pendapatan <=50K salah diprediksi sebagai >50K.
* **False Negative (FN)**: 284 individu dengan pendapatan >50K salah diprediksi sebagai <=50K.
* **True Positive (TP)**: 500 individu dengan pendapatan >50K diprediksi dengan benar.

Model menunjukkan kemampuan yang baik dalam mengklasifikasikan kelas mayoritas (<=50K), namun memiliki recall yang lebih rendah untuk kelas minoritas (>50K) (Recall untuk >50K = 500 / (500+284) = 0.6377 ~ 64%).

#### 5.3.4 Feature Importance

Berdasarkan output best\_model.feature\_importances\_ dari notebook (yang divisualisasikan dalam feature\_importance.png), fitur-fitur yang paling signifikan dalam menentukan prediksi pendapatan diidentifikasi. Urutan dan nilai kontribusi spesifiknya dapat dilihat pada visualisasi tersebut. Secara umum, fitur-fitur seperti marital\_status, capital\_gain, education\_num, age, dan hours\_per\_week diharapkan menunjukkan kontribusi penting.

### 5.4 Analisis Model

#### 5.4.1 Model Interpretability

Visualisasi decision tree (hingga kedalaman 3, seperti pada decision\_tree.png) memberikan pemahaman tentang bagaimana model membuat keputusan. Model ini bekerja berdasarkan serangkaian aturan if-then-else pada fitur-fitur input untuk mencapai prediksi kelas. Keterbacaan ini adalah salah satu keunggulan utama dari algoritma Decision Tree, memungkinkan pengguna untuk memahami logika di balik prediksi.

#### 5.4.2 Overfitting Analysis

Performa model pada training set (Akurasi: 86.50%) dan testing set (Akurasi: 86.08%) relatif dekat. Perbedaan yang kecil ini (sekitar 0.42%) menunjukkan bahwa model tidak mengalami overfitting yang signifikan dan memiliki kemampuan generalisasi yang baik pada data yang belum pernah dilihat sebelumnya. Hyperparameter seperti max\_depth, min\_samples\_leaf, dan min\_samples\_split yang dioptimalkan melalui GridSearchCV membantu mengontrol kompleksitas model dan mencegah overfitting.

#### 5.4.3 Feature Contribution

Analisis feature importance (seperti yang dibahas di 5.3.4 dan divisualisasikan di feature\_importance.png) mengindikasikan bahwa status perkawinan (marital\_status), perolehan modal (capital\_gain), tingkat pendidikan dalam tahun (education\_num), usia (age), dan jam kerja per minggu (hours\_per\_week) adalah beberapa faktor kunci dalam memprediksi tingkat pendapatan. Insight ini sejalan dengan pemahaman umum ekonomi bahwa status pernikahan (terutama Married-civ-spouse yang sering berkorelasi dengan pendapatan lebih tinggi), investasi/modal, tingkat pendidikan formal, pengalaman kerja (yang sering berkorelasi dengan usia), dan intensitas kerja merupakan determinan penting dari potensi pendapatan individu.

### 5.5 Hasil Implementasi Web Application

#### 5.5.1 User Interface

Aplikasi web Streamlit yang dikembangkan menyediakan antarmuka pengguna yang bersih dan intuitif untuk prediksi pendapatan. Pengguna dapat memasukkan nilai untuk 13 fitur demografis dan pekerjaan melalui sebuah form input yang terorganisir dalam beberapa bagian (Demographic, Work, Financial Information). Terdapat tombol “Predict” untuk memicu prediksi, “Reset” untuk mengembalikan semua field input ke nilai default, dan “Export Last Result” untuk mengaktifkan unduhan hasil prediksi terakhir. Hasil prediksi ditampilkan dengan jelas di kolom sebelah kanan, mencakup label kelas pendapatan yang diprediksi (‘>50K’ atau ‘<=50K’) dengan warna yang sesuai dan tingkat kepercayaan (confidence level) yang divisualisasikan menggunakan gauge chart interaktif dari Plotly. Selain itu, distribusi probabilitas untuk kedua kelas pendapatan juga ditampilkan dalam bentuk bar chart Plotly. Bagian akhir dari aplikasi menampilkan visualisasi horizontal bar chart dari feature importance model Decision Tree yang digunakan, juga menggunakan Plotly.

#### 5.5.2 Functionality Testing

Fungsionalitas aplikasi telah diuji dengan berbagai skenario input. Validasi input diimplementasikan untuk fitur-fitur numerik kunci (seperti age, education\_num, hours\_per\_week, capital\_gain, capital\_loss, fnlwgt) untuk memastikan bahwa data yang dimasukkan berada dalam rentang yang logis dan sesuai dengan batasan yang diobservasi dari dataset (sebagaimana didefinisikan dalam fungsi validate\_inputs). Jika input tidak valid, pesan error akan ditampilkan kepada pengguna di kolom hasil. Fungsi prediksi, reset input, dan ekspor hasil prediksi dalam format JSON telah diverifikasi dan berjalan sesuai dengan yang diharapkan. Prediksi hanya dijalankan jika tidak ada error validasi.

#### 5.5.3 Performance Analysis

Aplikasi web menunjukkan responsivitas yang baik terhadap interaksi pengguna. Pemuatan model dan komponen terkaitnya (seperti encoding maps dan feature names dari file income\_prediction\_components.joblib) dioptimalkan dengan penggunaan mekanisme caching (@st.cache\_resource pada Streamlit), yang memastikan model hanya dimuat sekali saat aplikasi pertama kali dijalankan atau saat ada perubahan pada file model. Hal ini mengurangi latensi pada prediksi berikutnya. Penggunaan visualisasi interaktif dari Plotly untuk menampilkan hasil prediksi dan feature importance juga meningkatkan pengalaman pengguna, membuat informasi lebih mudah dipahami dan diinterpretasi.

## 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

1. **Model Performance**: Model Decision Tree yang dikembangkan, setelah optimasi hyperparameter (criterion='gini', max\_depth=10, min\_samples\_leaf=4, min\_samples\_split=10), berhasil mencapai akurasi **86.08%** pada data testing untuk prediksi tingkat pendapatan. Feature importance tertinggi teridentifikasi pada fitur-fitur seperti **marital\_status, capital\_gain, dan education\_num**.
2. **Feature Analysis**: Fitur yang paling berpengaruh terhadap prediksi pendapatan, berdasarkan analisis feature importance, adalah **marital\_status, capital\_gain, education\_num, age, dan hours\_per\_week**. Temuan ini konsisten dengan teori dan intuisi ekonomi mengenai faktor-faktor penentu pendapatan.
3. **Implementation Success**: Penelitian ini berhasil mengimplementasikan end-to-end machine learning pipeline, mulai dari akuisisi dan pembersihan data, eksplorasi data, feature engineering, training model, evaluasi, hingga deployment model sebagai aplikasi web interaktif menggunakan Streamlit.
4. **Business Value**: Model yang dikembangkan dapat berfungsi sebagai alat bantu dalam pengambilan keputusan atau analisis untuk berbagai stakeholder, dengan tingkat akurasi yang cukup baik untuk aplikasi di dunia nyata, khususnya sebagai alat skrining awal atau untuk mendapatkan insight dari data demografis terkait potensi pendapatan.

### 6.2 Kontribusi Penelitian

1. **Metodologi**: Demonstrasi penerapan best practices dalam machine learning pipeline, termasuk EDA, preprocessing (penanganan missing values, encoding), hyperparameter tuning dengan cross-validation, dan evaluasi model yang komprehensif menggunakan berbagai metrik.
2. **Implementasi**: Menyediakan contoh konkret deployment model machine learning sebagai aplikasi web yang user-friendly menggunakan Streamlit, menjembatani kesenjangan antara model analitik dan pengguna akhir dengan fitur seperti input validasi, visualisasi interaktif, dan ekspor hasil.
3. **Interpretability**: Melakukan analisis mendalam terhadap feature importance dan memungkinkan visualisasi Decision Tree (meskipun tidak secara langsung di app) untuk memberikan insight bisnis yang dapat dipahami dan ditindaklanjuti dari model yang relatif transparan.

### 6.3 Limitasi Penelitian

1. **Data Currency**: Dataset yang digunakan berasal dari tahun 1994. Kondisi sosial-ekonomi dan pasar kerja mungkin telah banyak berubah, sehingga relevansi model untuk prediksi kondisi saat ini bisa terbatas.
2. **Algorithm Scope**: Penelitian ini hanya berfokus pada algoritma Decision Tree. Perbandingan dengan algoritma klasifikasi lain (misalnya, Random Forest, Gradient Boosting, SVM, atau Neural Networks) tidak dilakukan, yang mungkin bisa memberikan performa lebih baik atau insight berbeda.
3. **Feature Engineering**: Meskipun feature engineering dasar telah dilakukan (seperti encoding dan penghapusan fitur redundan), eksplorasi feature engineering yang lebih sophisticated (misalnya, pembuatan fitur interaksi, binning fitur numerik secara lebih mendalam) mungkin dapat lebih meningkatkan performa model.
4. **Missing Value Imputation**: Imputasi modus digunakan untuk menangani missing values pada fitur kategorikal. Metode imputasi yang lebih canggih (misalnya, k-NN imputer atau model-based imputation) bisa dieksplorasi.
5. **Imbalance Target Class**: Kelas target income tidak seimbang (sekitar 76% <=50K dan 24% >50K). Meskipun model mencapai akurasi yang baik, performa pada kelas minoritas (recall 64% untuk >50K) lebih rendah. Teknik penanganan data tidak seimbang (misalnya, SMOTE, ADASYN, atau penggunaan class weights) tidak secara eksplisit diterapkan dalam training model (selain stratify pada train\_test\_split).

### 6.4 Saran untuk Penelitian Selanjutnya

1. **Algorithm Comparison**: Melakukan studi komparatif dengan algoritma klasifikasi lain yang lebih canggih seperti Random Forest, XGBoost, LightGBM, atau Neural Networks untuk melihat potensi peningkatan akurasi dan metrik lainnya, terutama recall untuk kelas minoritas.
2. **Advanced Feature Engineering**: Mengeksplorasi teknik feature engineering yang lebih lanjut, seperti polynomial features, interaction terms antar fitur yang signifikan secara domain, atau binning fitur numerik (misalnya age, hours\_per\_week) menjadi kategori-kategori yang lebih bermakna.
3. **Handling Imbalanced Data**: Mengimplementasikan teknik khusus untuk menangani ketidakseimbangan kelas target secara eksplisit, seperti oversampling (misalnya SMOTE), undersampling, atau menggunakan parameter class\_weight='balanced' pada algoritma model yang mendukung, untuk meningkatkan prediksi pada kelas minoritas.
4. **Updated Dataset**: Menggunakan dataset yang lebih baru dan relevan dengan kondisi sosial-ekonomi terkini untuk meningkatkan daya generalisasi dan relevansi praktis model.
5. **Model Explainability (XAI)**: Lebih mendalami aspek explainability model menggunakan teknik seperti SHAP (SHapley Additive exPlanations) atau LIME (Local Interpretable Model-agnostic Explanations), terutama jika di masa depan menggunakan model yang lebih kompleks (black-box).
6. **Deployment Enhancement**: Meningkatkan aplikasi web dengan fitur tambahan seperti kemampuan untuk batch prediction (mengunggah file data), penyimpanan histori prediksi pengguna, atau integrasi dengan dashboard monitoring performa model secara berkala dan mekanisme retraining otomatis.
7. **Cross-Cultural Analysis**: Jika data tersedia, melakukan analisis serupa pada dataset dari negara atau konteks budaya yang berbeda untuk membandingkan faktor-faktor penentu pendapatan dan universalitas model.

## DAFTAR PUSTAKA

1. Dua, S., & Du, X. (2016). Data mining and machine learning in cybersecurity. CRC press.
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
3. Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In Proceedings of the second international conference on knowledge discovery and data mining (pp. 202-207).
4. UCI Machine Learning Repository. Adult Data Set. https://archive.ics.uci.edu/dataset/2/adult
5. [Tambahkan referensi lain yang relevan dengan topik income prediction, decision tree, dan machine learning]

## LAMPIRAN

### Lampiran A: Source Code

[Link ke repository GitHub atau lampiran kode lengkap dari klasifikasi.ipynb dan streamlit\_income\_app.py]

### Lampiran B: Dataset Description

[Detail lengkap tentang dataset dari sumber UCI atau deskripsi yang lebih rinci]

### Lampiran C: Additional Visualizations

[Visualisasi tambahan yang tidak masuk ke bagian utama, seperti semua distribution\_[feature].png, correlation\_matrix.png, correlation\_after\_encoding.png, final\_correlation.png, decision\_tree.png (full depth jika ada), feature\_importance.png]

### Lampiran D: Model Artifacts

[Informasi tentang model yang disimpan (income\_prediction\_components.joblib) dan cara penggunaannya. Ini mencakup detail tentang struktur dictionary yang disimpan: model (objek DecisionTreeClassifier), feature\_names (list nama fitur input), encoding\_maps (dictionary berisi mapping untuk setiap fitur kategorikal dan target), model\_params (parameter terbaik dari GridSearchCV), removed\_features (list fitur yang dihapus), target\_map (mapping untuk variabel target).]