Analisis Pengaruh Tingkat Pendidikan terhadap Tingkat Pengangguran di Indonesia Menggunakan K-Means dan Decision Tree

DISUSUN OLEH

KELOMPOK 8 :

M. Hafiz Andrean Siregar	102022300045
Muhammad Galih Ilham	102022300364
Ghifari Derriel aryasatya	102022330310

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS REKAYASA INDUSTRI

UNIVERSITAS TELKOM

BANDUNG

Kata Pengantar

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa karena atas limpahan rahmat dan karunia-Nya, kami dapat menyelesaikan laporan tugas besar mata kuliah Data Mining dengan judul "Analisis Pengaruh Tingkat Pendidikan terhadap Tingkat Pengangguran di Indonesia Menggunakan K-Means dan Decision Tree." Laporan ini disusun sebagai bentuk implementasi dari teori dan praktik yang telah kami pelajari selama perkuliahan, khususnya dalam mengaplikasikan teknik data mining untuk memecahkan permasalahan sosial yang relevan.

Permasalahan pengangguran yang masih tinggi di Indonesia menjadi perhatian utama dalam penelitian ini. Kami melihat adanya kebutuhan untuk mengevaluasi kembali sejauh mana pendidikan mampu menjadi faktor penentu dalam penyerapan tenaga kerja. Dengan latar belakang tersebut, kami mencoba menganalisis data pengangguran berdasarkan tingkat pendidikan menggunakan dua metode yang populer dalam data mining, yaitu K-Means Clustering dan Decision Tree Classification. Pendekatan ini diharapkan dapat memberikan sudut pandang baru dalam memahami permasalahan pengangguran dari sisi pendidikan.

Dalam penyusunan laporan ini, kami menghadapi berbagai tantangan, baik dalam hal teknis pengolahan data maupun dalam menginterpretasikan hasil analisis. Namun, dengan kerja sama tim dan bimbingan dari dosen pengampu serta referensi dari berbagai sumber, kami berhasil menyelesaikan tugas ini sesuai dengan harapan. Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran sangat kami harapkan demi penyempurnaan di masa mendatang.

Akhir kata, kami mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung, dalam proses penyusunan laporan ini. Semoga laporan ini dapat memberikan kontribusi positif dan menjadi referensi yang bermanfaat bagi semua pihak yang berkepentingan.

Daftar Isi

Daftar Gambar

Daftar Tabel

Ringkasan

Tingkat pengangguran merupakan indikator utama dalam menilai kesejahteraan dan perkembangan ekonomi suatu negara. Di Indonesia, meskipun tingkat pendidikan masyarakat mengalami peningkatan dari tahun ke tahun, fenomena pengangguran justru masih menjadi permasalahan signifikan. Bahkan, sejumlah lulusan pendidikan tinggi pun mengalami

kesulitan dalam memperoleh pekerjaan yang sesuai dengan kompetensinya. Oleh karena itu, penting untuk meneliti lebih jauh bagaimana hubungan antara tingkat pendidikan dan tingkat pengangguran di Indonesia.

Penelitian ini dilakukan dengan pendekatan kuantitatif eksploratif menggunakan algoritma K-Means Clustering dan Decision Tree. Data yang digunakan merupakan data sekunder dari sumber resmi seperti Badan Pusat Statistik (BPS) yang memuat informasi jumlah pengangguran dan tingkat pendidikan penduduk Indonesia. K-Means digunakan untuk mengelompokkan data ke dalam beberapa klaster berdasarkan kesamaan karakteristiknya, sedangkan Decision Tree digunakan untuk membangun model klasifikasi yang memprediksi status pengangguran berdasarkan variabel pendidikan.

Hasil analisis menunjukkan bahwa terdapat kelompok tertentu, khususnya lulusan Sekolah Menengah Kejuruan (SMK) dan Sekolah Menengah Atas (SMA), yang memiliki tingkat pengangguran relatif lebih tinggi dibandingkan kelompok lainnya. K-Means berhasil membentuk klaster yang menggambarkan sebaran kelompok pengangguran berdasarkan jenjang pendidikan. Sementara itu, model Decision Tree mampu mengidentifikasi variabel pendidikan sebagai atribut yang sangat signifikan dalam menentukan apakah seseorang termasuk dalam kelompok pengangguran.

Dengan demikian, hasil penelitian ini memberikan insight penting bahwa peningkatan pendidikan saja tidak cukup jika tidak diimbangi dengan relevansi kurikulum dan kebutuhan pasar kerja. Pemerintah dan lembaga pendidikan diharapkan dapat merumuskan kebijakan yang lebih tepat sasaran untuk mengurangi angka pengangguran, khususnya dengan meningkatkan keterampilan kerja praktis dan konektivitas dengan dunia industri.

Kata Kunci: Pengangguran, Tingkat Pendidikan, K-Means, Decision Tree, Data Mining, Indonesia.

Bab I

Pendahuluan

1.1 Latar Belakang

Tingkat pengangguran merupakan salah satu indikator utama dalam mengukur stabilitas ekonomi dan kesejahteraan sosial suatu negara. Di Indonesia, persoalan pengangguran masih

menjadi tantangan besar, terutama ketika dikaitkan dengan tingkat pendidikan masyarakat. Idealnya, semakin tinggi tingkat pendidikan seseorang, maka semakin besar pula peluangnya untuk memperoleh pekerjaan. Namun pada kenyataannya, banyak lulusan dari berbagai jenjang pendidikan termasuk perguruan tinggi yang masih mengalami kesulitan dalam mendapatkan pekerjaan.

Fenomena tersebut menunjukkan adanya ketidakseimbangan antara kualitas dan kuantitas tenaga kerja terdidik dengan kebutuhan aktual dari dunia industri. Permasalahan ini dapat disebabkan oleh berbagai faktor, di antaranya ketidaksesuaian kurikulum dengan kebutuhan dunia kerja (job mismatch), rendahnya keterampilan praktis lulusan, hingga terbatasnya lapangan pekerjaan di sektor formal. Oleh karena itu, diperlukan suatu pendekatan analitis berbasis data untuk mengevaluasi dan memahami bagaimana tingkat pendidikan memengaruhi tingkat pengangguran secara kuantitatif dan objektif.

Dengan kemajuan teknologi informasi dan tersedianya data statistik ketenagakerjaan, teknik data mining dapat dimanfaatkan untuk menganalisis hubungan antara pendidikan dan pengangguran. Dalam penelitian ini, dilakukan perhitungan rata-rata tingkat pendidikan penduduk per provinsi berdasarkan distribusi jenjang pendidikan, serta simulasi tingkat pengangguran sebagai fungsi eksponensial dari variabel pendidikan. Melalui metode K-Means Clustering, provinsi-provinsi di Indonesia dikelompokkan berdasarkan karakteristik pendidikan dan pengangguran. Selain itu, digunakan Decision Tree Regression untuk memetakan hubungan prediktif, dan Regresi Linear untuk melihat kekuatan hubungan linier antara kedua variabel.

Pendekatan ini diharapkan mampu memberikan gambaran yang lebih dalam mengenai keterkaitan antara pendidikan dan pengangguran, serta menghasilkan rekomendasi kebijakan yang lebih tepat guna dalam rangka meningkatkan efektivitas pendidikan dan mengurangi tingkat pengangguran secara nasional.

1.2 Rumusan Masalah

Permasalahan yang diangkat dalam penelitian ini adalah:

- Bagaimana pola hubungan antara tingkat pendidikan dengan tingkat pengangguran di Indonesia?
- Bagaimana segmentasi dan klasifikasi data pengangguran dapat dilakukan menggunakan algoritma *K-Means* dan *Decision Tree*?

Solusi yang ditawarkan adalah dengan menerapkan metode *K-Means* untuk mengelompokkan data berdasarkan karakteristik tertentu dan *Decision Tree* untuk mengidentifikasi pengaruh variabel pendidikan terhadap status pengangguran.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan model analisis berbasis data mining yang mampu mengungkap hubungan antara rata-rata tingkat pendidikan dan tingkat pengangguran di Indonesia. Model ini tidak hanya bertujuan untuk mengidentifikasi pola dan segmentasi wilayah berdasarkan kedua indikator tersebut, tetapi juga untuk membangun model prediktif yang dapat memberikan wawasan strategis dalam perumusan kebijakan publik.

Secara khusus, tujuan penelitian ini adalah sebagai berikut:

1. Menghitung rata-rata lama sekolah (Pendidikan_RataRata) berdasarkan distribusi jenjang pendidikan penduduk di setiap provinsi.

2. Mengelompokkan provinsi-provinsi di Indonesia ke dalam klaster berdasarkan variabel pendidikan dan pengangguran menggunakan metode K-Means Clustering.

3. Membangun model klasifikasi dan prediksi menggunakan algoritma Decision Tree Regression untuk memetakan pola keterkaitan pendidikan terhadap pengangguran.

4. Mengkaji hubungan linier antara rata-rata pendidikan dan tingkat pengangguran dengan model Regresi Linear, serta mengevaluasi kinerja model melalui metrik RMSE dan R^2.

5. Menyajikan hasil analisis dalam bentuk visualisasi dan dashboard interaktif yang informatif bagi pembaca maupun pemangku kepentingan.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini dapat diklasifikasikan ke dalam dua kategori, yaitu manfaat teoretis dan manfaat praktis:

1. **Manfaat Teoretis**

- Memberikan kontribusi terhadap pengembangan ilmu pengetahuan, khususnya dalam penerapan teknik data mining untuk menganalisis fenomena sosial-ekonomi.

- Menjadi referensi metodologis dalam penggunaan algoritma K-Means Clustering, Decision Tree, dan Linear Regression pada data statistik ketenagakerjaan.

- Meningkatkan pemahaman akademisi dan mahasiswa dalam membangun pipeline analisis data mulai dari eksplorasi, transformasi, hingga modeling.

2. **Manfaat Praktis**

- Memberikan informasi berbasis data kepada pemerintah pusat dan daerah dalam merumuskan kebijakan pendidikan dan ketenagakerjaan.

- Menjadi alat bantu dalam mengidentifikasi kelompok wilayah yang memiliki tingkat pendidikan rendah dan tingkat pengangguran tinggi.

- Mendorong evaluasi kebijakan pendidikan nasional agar lebih terfokus pada peningkatan kualitas lulusan yang sesuai dengan kebutuhan pasar kerja.

- Menjadi contoh studi kasus penerapan analitik data di bidang sosial-ekonomi untuk mahasiswa dan praktisi bidang sistem informasi, data science, serta perencanaan pembangunan.

1.5 Batasan Penelitian

Untuk menjaga fokus dan cakupan analisis agar tetap terkendali serta relevan terhadap tujuan penelitian, maka penelitian ini memiliki beberapa batasan sebagai berikut:

1. Penelitian ini hanya menganalisis data agregat per provinsi di Indonesia, tanpa memperhitungkan variasi individual seperti jenis kelamin, usia, atau kondisi ekonomi rumah tangga.

2. Data yang digunakan bersumber dari dataset sekunder yang bersifat kuantitatif, khususnya data distribusi tingkat pendidikan dan jumlah pengangguran per tahun/periode.

3. Simulasi tingkat pengangguran menggunakan pendekatan fungsi eksponensial terhadap rata-rata pendidikan dengan penambahan noise acak untuk menyesuaikan variasi realistik.

4. Analisis model terbatas pada tiga algoritma utama: K-Means untuk segmentasi, Decision Tree Regression untuk klasifikasi regresif, dan Regresi Linear untuk evaluasi hubungan linier.

5. Hasil akhir penelitian ini disajikan dalam bentuk visualisasi grafik dan dashboard, tanpa dilakukan pengembangan aplikasi sistem informasi terintegrasi atau web-based secara penuh.

Bab II

Landasan Teori/ Tinjauan Pustaka

2.1 Data Mining

Data mining adalah proses penemuan pola, hubungan, tren, dan informasi penting dari kumpulan data besar menggunakan teknik statistik, matematika, dan pembelajaran mesin. Proses ini memungkinkan penggalian informasi tersembunyi yang berguna untuk pengambilan keputusan berbasis data. Dalam konteks pengaruh pendidikan terhadap pengangguran, data mining dapat mengidentifikasi pola-pola yang tidak tampak secara kasat mata dari data pendidikan dan ketenagakerjaan.

Menurut Han dan Kamber (2011), proses data mining meliputi pembersihan data, integrasi data, seleksi data, transformasi data, penambangan pola, evaluasi pola, dan penyajian

informasi. Proses ini sangat penting untuk mengelola data besar dari sumber resmi seperti BPS sehingga dapat diolah menjadi insight yang bermanfaat.

2.2 K-Means Clustering

K-Means adalah algoritma clustering yang digunakan untuk mengelompokkan data ke dalam sejumlah kelompok (klaster) berdasarkan jarak rata-rata (centroid). Algoritma ini bekerja dengan membagi data menjadi k kelompok sehingga setiap data berada pada klaster dengan centroid terdekat.

Dalam studi ini, K-Means digunakan untuk mengelompokkan wilayah atau individu berdasarkan tingkat pendidikan dan tingkat pengangguran yang dimiliki. Dengan teknik ini, kelompok yang memiliki karakteristik pendidikan dan pengangguran serupa dapat diidentifikasi.

K-Means sangat berguna dalam segmentasi data dan memudahkan analisis selanjutnya untuk pengambilan keputusan yang tepat.

2.3 Decision Tree

Decision Tree adalah metode pembelajaran mesin yang digunakan untuk klasifikasi dan prediksi berdasarkan serangkaian aturan keputusan yang terbentuk dalam bentuk pohon. Model ini mudah dipahami karena berbentuk diagram yang menunjukkan cabang keputusan dari atribut input ke hasil prediksi.

Dalam konteks penelitian ini, Decision Tree digunakan untuk memprediksi status pengangguran berdasarkan variabel tingkat pendidikan dan faktor-faktor lain yang relevan. Model ini membantu mengidentifikasi faktor-faktor determinan utama yang mempengaruhi pengangguran.

Keunggulan Decision Tree adalah kemampuannya memberikan aturan yang jelas untuk interpretasi dan keputusan.

2.4 Linear Regression

Regresi linier adalah metode statistik yang digunakan untuk memodelkan hubungan antara satu variabel dependen dan satu atau lebih variabel independen dengan garis lurus. Tujuan utamanya adalah untuk memprediksi nilai variabel dependen berdasarkan variabel independen.

Dalam penelitian ini, regresi linier digunakan untuk menganalisa dan memprediksi pengaruh tingkat pendidikan terhadap tingkat pengangguran secara kuantitatif. Model ini membantu mengukur seberapa besar perubahan tingkat pendidikan berdampak pada tingkat pengangguran.

Regresi linier sangat efektif dalam menentukan tren dan kekuatan hubungan antar variabel.

2.5 Sistem Informasi dan Visualisasi Data

Sistem informasi berperan penting dalam pengelolaan dan analisis data ketenagakerjaan dan pendidikan. Visualisasi data melalui grafik, diagram, dan dashboard interaktif memudahkan pemahaman hasil analisis.

Dashboard interaktif memungkinkan pengguna untuk memantau tren pengangguran dan pendidikan secara real-time dan mengeksplorasi data secara intuitif.

Teknologi visualisasi seperti Power BI, Tableau, atau platform berbasis web memudahkan penyampaian insight kepada pengambil kebijakan dan masyarakat luas.

2.6 Studi Terdahulu

Berikut adalah beberapa studi terdahulu yang relevan:

- "Analysis of Education Impact on Unemployment Rate Using Machine Learning Techniques" oleh Santoso et al. (2020) mengkombinasikan clustering dan regresi untuk menganalisis pengaruh pendidikan terhadap pengangguran di Indonesia dengan hasil signifikan.

- "Predicting Unemployment Status Using Decision Tree and Logistic Regression" oleh Wijaya dan Lestari (2021) menunjukkan bahwa Decision Tree efektif dalam mengidentifikasi faktor utama pengangguran di kalangan lulusan sekolah menengah.

- "Spatial and Statistical Analysis of Education and Employment Data" oleh Rahman et al. (2019) menekankan pentingnya pemetaan dan visualisasi data untuk pengambilan keputusan kebijakan ketenagakerjaan.

Penelitian-penelitian tersebut mendukung penggunaan metode yang dipilih dalam studi ini, menunjukkan efektivitas teknik clustering, decision tree, dan regresi dalam konteks pengangguran dan pendidikan.

Bab III

Metodologi Penelitian

3.1 Jenis Penelitian dan Dataset yang Digunakan

Penelitian ini termasuk ke dalam penelitian kuantitatif karena menggunakan data numerik berupa tingkat pendidikan, tingkat pengangguran, dan variabel demografis lain yang diolah menggunakan metode statistik dan algoritma machine learning.

Dataset yang digunakan terdiri dari beberapa sumber utama:

- **Data Tingkat Pendidikan dan Pengangguran di Indonesia (2015–2022):** Data ini diperoleh dari publikasi resmi Badan Pusat Statistik (BPS) yang mencakup tingkat pendidikan formal, jumlah angkatan kerja, dan tingkat pengangguran terbuka di tingkat provinsi dan kabupaten/kota.

- **Data Demografis Pendukung:** Variabel tambahan seperti usia, jenis kelamin, dan lokasi geografis dari sumber BPS untuk memperkaya analisis.

3.2 Sistematika Penyelesaian Permasalahan

Tahapan metode yang digunakan dalam penelitian ini adalah sebagai berikut:

1. **Pengumpulan dan Integrasi Data**
    - Mengumpulkan data dari berbagai publikasi BPS dan sumber resmi terkait.
    - Mengintegrasikan data menjadi satu database terpadu dengan format seragam.
2. **Eksplorasi dan Pembersihan Data**
    - Melakukan eksplorasi awal untuk memahami distribusi dan pola data.
    - Membersihkan data dengan menghapus nilai yang hilang (missing values) dan data duplikat.
3. **Transformasi dan Normalisasi Data**
    - Menambah atribut baru seperti rasio pendidikan terhadap jumlah angkatan kerja.
    - Melakukan normalisasi agar atribut dengan skala berbeda dapat diproses secara seimbang.
4. **Penerapan Clustering dengan K-Means**
    - Mengelompokkan data wilayah atau individu berdasarkan karakteristik pendidikan dan pengangguran.
5. **Pembangunan Model Prediksi dengan Decision Tree dan Linear Regression**
    - Decision Tree digunakan untuk klasifikasi status pengangguran berdasarkan variabel pendidikan dan demografis.
    - Linear Regression digunakan untuk memprediksi tingkat pengangguran sebagai variabel kontinu berdasarkan tingkat pendidikan dan faktor lain.
6. **Evaluasi Model dan Validasi Hasil**

    - Melakukan evaluasi model clustering dan prediksi menggunakan metrik evaluasi yang sesuai.

7. **Visualisasi Data dan Hasil Analisis**
    - Membuat dashboard interaktif untuk memudahkan interpretasi dan pengambilan keputusan.

3.3 Detail Data Exploration

Eksplorasi data yang dilakukan meliputi:

- Analisis distribusi tingkat pendidikan di berbagai wilayah.
- Distribusi tingkat pengangguran menurut kelompok umur dan jenis kelamin.
- Korelasi awal antara tingkat pendidikan dan pengangguran.
- Visualisasi geografis distribusi pendidikan dan pengangguran menggunakan peta tematik dan heatmap.

3.4 Detail Data Preparation

Tahapan data preparation terdiri dari:

- **Pembersihan Data:** Menghapus data yang kosong atau tidak lengkap, menghapus kolom yang tidak relevan.
- **Transformasi Data:** Membuat atribut baru seperti persentase lulusan SMA dan perguruan tinggi di setiap wilayah.
- **Normalisasi:** Menggunakan Min-Max Scaling agar variabel dengan rentang berbeda tidak mendominasi hasil clustering dan prediksi.
- **Penggabungan Data:** Menggabungkan data pendidikan, pengangguran, dan variabel demografis ke dalam satu dataset yang siap dianalisis.

3.5 Detail Modelling dan Evaluasi

Clustering

- **Algoritma:** K-Means dengan penentuan jumlah klaster optimal menggunakan metode Elbow dan Silhouette Score.

- **Input:** Variabel tingkat pendidikan, tingkat pengangguran, dan variabel demografis terkait.

- **Evaluasi:**

    - Silhouette Score untuk menilai kualitas klaster.

    - Visualisasi hasil klaster pada peta atau diagram untuk memudahkan interpretasi.

Prediksi

- **Decision Tree:**

    - Input: Variabel tingkat pendidikan, usia, jenis kelamin, dan data pendukung lain.

    - Target: Status pengangguran (menganggur atau bekerja).

    - Evaluasi: Akurasi, Precision, Recall, dan F1-Score.

- **Linear Regression:**

    - Input: Variabel tingkat pendidikan dan faktor pendukung.

    - Target: Tingkat pengangguran (variabel kontinu).

    - Evaluasi: Koefisien determinasi (R^2), Root Mean Square Error (RMSE).

3.6 Detail Perancangan Dashboard

Dashboard penelitian ini dirancang untuk menampilkan:

- **Peta Klaster:** Visualisasi hasil clustering K-Means dalam bentuk peta interaktif menggunakan Leaflet atau Python Folium yang menampilkan wilayah dengan karakteristik pendidikan dan pengangguran berbeda.

- **Grafik Tren:** Visualisasi grafik tren historis tingkat pendidikan dan pengangguran dari tahun ke tahun.

- **Hasil Prediksi:** Peta dan grafik hasil prediksi dari model Decision Tree dan Linear Regression.

- **Fitur Interaktif:**

    - Filter berdasarkan provinsi, kabupaten/kota, dan rentang tahun.

- Slider waktu untuk melihat perubahan data dari waktu ke waktu.
- Tooltip untuk menampilkan informasi detail wilayah saat pointer diarahkan ke peta.

Bab IV

Hasil dan Pembahasan

Data Exploration

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.tree import DecisionTreeRegressor, plot_tree
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('dataset_tubes.csv')
df.head()
```

[1] ✓ 1.6s Python

	Periode	Bulan	Tidak/belum pernah sekolah	Tidak/belum tamat SD	SD	SLTP	SLTA Umum/SMU	SLTA Kejuruan/SMK	Akademi/Diploma	Universitas	Total
0	2006	Februari	234465	614960	2675459	2860007	2842876	1204140	297185	375601	11104693
1	2006	Agustus	170666	611254	2589699	2730045	2851518	1305190	278074	395554	10932000
2	2007	Februari	145750	520316	2753548	2643062	2630360	1114675	330316	409890	10547917
3	2007	Agustus	94301	438519	2179792	2264198	2532204	1538349	397191	566588	10011142
4	2008	Februari	79764	448431	2216748	2166619	2204377	1165582	519867	626202	9427590

Gambar 4.1: *Data Exploration*

Data Preparation

Exploratory Data Analysis

```python
# Asumsi lama tahun pendidikan
tahun_pendidikan = {
    'Tidak/belum pernah sekolah': 0,
    'Tidak/belum tamat SD': 3,
    'SD': 6,
    'SLTP': 9,
    'SLTA Umum/SMU': 12,
    'SLTA Kejuruan/SMK': 12,
    'Akademi/Diploma': 14,
    'Universitas': 16
}

# Hitung total populasi berpendidikan dan jumlah tahun total
df['Total_Pendidikan_Terpenuhi'] = (
    df['Tidak/belum pernah sekolah'] * tahun_pendidikan['Tidak/belum pernah sekolah'] +
    df['Tidak/belum tamat SD'] * tahun_pendidikan['Tidak/belum tamat SD'] +
    df['SD'] * tahun_pendidikan['SD'] +
    df['SLTP'] * tahun_pendidikan['SLTP'] +
    df['SLTA Umum/SMU'] * tahun_pendidikan['SLTA Umum/SMU'] +
    df['SLTA Kejuruan/SMK'] * tahun_pendidikan['SLTA Kejuruan/SMK'] +
    df['Akademi/Diploma'] * tahun_pendidikan['Akademi/Diploma'] +
    df['Universitas'] * tahun_pendidikan['Universitas']
)

df['Pendidikan_RataRata'] = df['Total_Pendidikan_Terpenuhi'] / df['Total']
df.head()
```

[2] ✓ 0.0s Python

	Periode	Bulan	Tidak/belum pernah sekolah	Tidak/belum tamat SD	SD	SLTP	SLTA Umum/SMU	SLTA Kejuruan/SMK	Akademi/Diploma	Universitas	Total	Total_Pendidikan_Terpenuhi	Pendidikan_RataRat
0	2006	Februari	234465	614960	2675459	2860007	2842876	1204140	297185	375601	11104693	102372095	9.21881
1	2006	Agustus	170666	611254	2589699	2730045	2851518	1305190	278074	395554	10932000	102044757	9.33450
2	2007	Februari	145750	520316	2753548	2643062	2630360	1114675	330316	409890	10547917	97992878	9.29025
3	2007	Agustus	94301	438519	2179792	2264198	2532204	1538349	397191	566588	10011142	98244809	9.81354
4	2008	Februari	79764	448431	2216748	2166619	2204377	1165582	519867	626202	9427590	91882230	9.74609

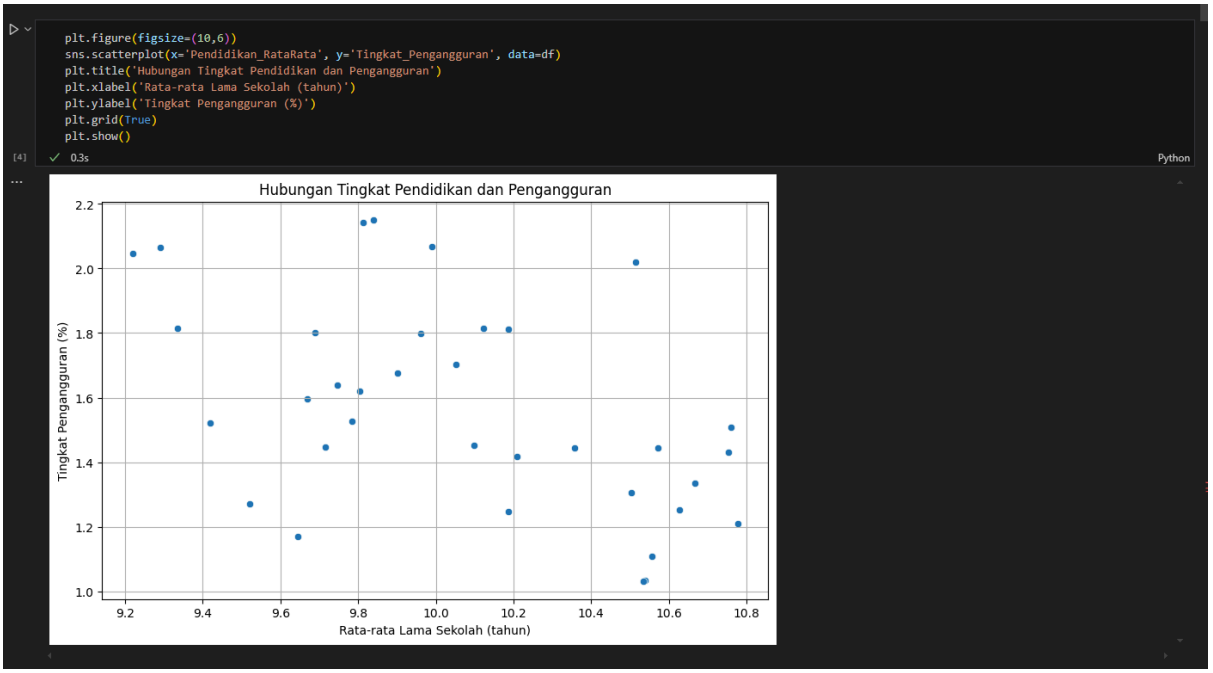
Gambar 4.2: *Data Preparation 1*

```python
# Simulasi: semakin tinggi rata-rata pendidikan, semakin rendah tingkat pengangguran
np.random.seed(42)
df['Tingkat_Pengangguran'] = 12 * np.exp(-0.2 * df['Pendidikan_RataRata']) + np.random.normal(0, 0.3, size=len(df))
df['Tingkat_Pengangguran'] = df['Tingkat_Pengangguran'].clip(lower=1)  # minimal 1%
df[['Periode', 'Pendidikan_RataRata', 'Tingkat_Pengangguran']].head()
```

[3] ✓ 0.0s Python

	Periode	Pendidikan_RataRata	Tingkat_Pengangguran
0	2006	9.218814	2.047666
1	2006	9.334500	1.813747
2	2007	9.290259	2.066021
3	2007	9.813547	2.142637
4	2008	9.746099	1.638375

Gambar 4.3: *Data Preparation 2*



Gambar 4.4: *Data Preparation 3*

Modelling dan evaluasi spasial K-Means clustering

K-Means Clustering

```
def find_outlier_boundary(df, variable):


    IQR = df[variable].quantile(0.75) - df[variable].quantile(0.25)

    lower_boundary = df[variable].quantile(0.25) - (IQR * 1.5)
    upper_boundary = df[variable].quantile(0.75) + (IQR * 1.5)

    return upper_boundary, lower_boundary
```
[5] ✓ 0.0s Python

```
full_occup_upper_limit, full_occup_lower_limit = find_outlier_boundary(df,'Universitas')
full_occup_upper_limit, full_occup_lower_limit
```
[6] ✓ 0.0s Python

```
(np.float64(1036088.5), np.float64(255484.5))
```

```
full_occup_upper_limit, full_occup_lower_limit = find_outlier_boundary(df,'Tidak/belum pernah sekolah')
full_occup_upper_limit, full_occup_lower_limit
```
[7] ✓ 0.0s Python

```
(np.float64(211656.5), np.float64(-65239.5))
```

```
data_clf = df[(df['Tidak/belum pernah sekolah'] <= full_occup_upper_limit) & (df['Tidak/belum pernah sekolah'] >= full_occup_lower_limit)]
```
[8] ✓ 0.0s Python

```
data_clf = df[(df['Tidak/belum pernah sekolah'] <= full_occup_upper_limit) & (df['Universitas'] >= full_occup_lower_limit)]
```
[9] ✓ 0.0s Python

Gambar 4. 5: *Modelling dan evaluasi spasial K-Means clustering 1*

```
print(data_clf.columns.tolist())
```
[10] ✓ 0.0s Python

```
['Periode', 'Bulan', 'Tidak/belum pernah sekolah', 'Tidak/belum tamat SD', 'SD', 'SLTP', 'SLTA Umum/SMU', 'SLTA Kejuruan/SMK', 'Akademi/Diploma', 'Universitas', 'Total', 'Tot
```

```
data_clf.head()
```
[11] ✓ 0.0s Python

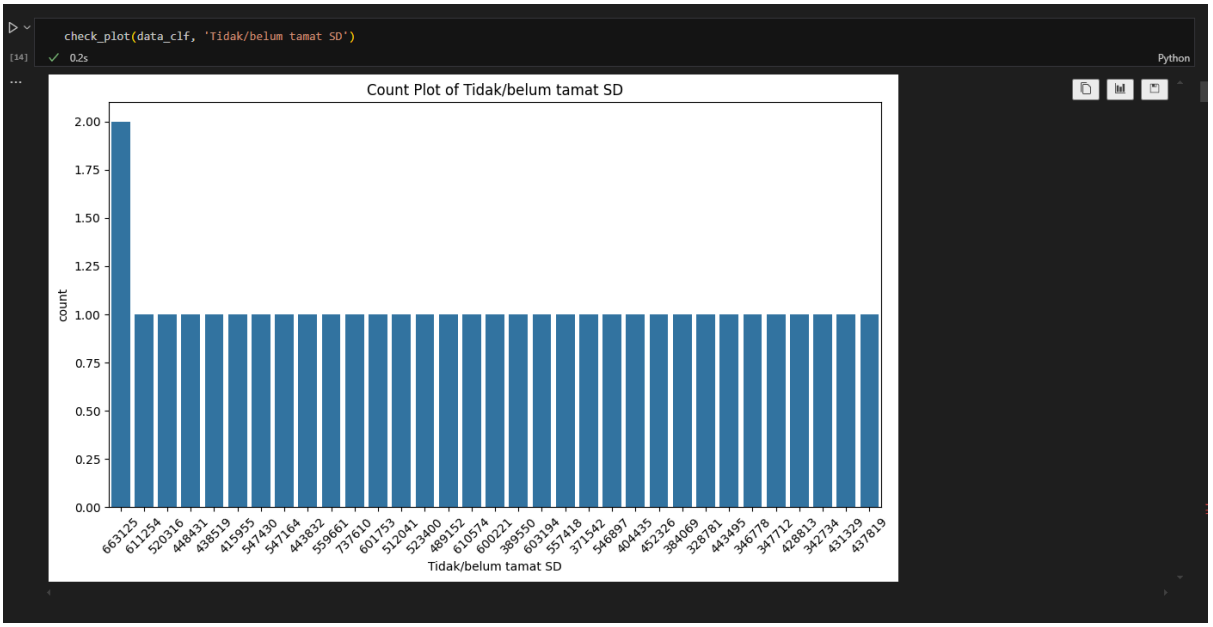
	Periode	Bulan	Tidak/belum pernah sekolah	Tidak/belum tamat SD	SD	SLTP	SLTA Umum/SMU	SLTA Kejuruan/SMK	Akademi/Diploma	Universitas	Total	Total_Pendidikan_Terpenuhi	Pendidikan_RataRat
1	2006	Agustus	170666	611254	2589699	2730045	2851518	1305190	278074	395554	10932000	102044757	9.33450
2	2007	Februari	145750	520316	2753548	2643062	2630360	1114675	330316	409890	10547917	97992878	9.29025
3	2007	Agustus	94301	438519	2179792	2264198	2532204	1538349	397191	566588	10011142	98244809	9.81354
4	2008	Februari	79764	448431	2216748	2166619	2204377	1165582	519867	626202	9427590	91882230	9.74609
5	2008	Agustus	103206	443832	2099968	1973986	2403394	1409128	362683	598318	9394515	92098092	9.80339

```
# Ubah nama kolom jika perlu
data_clf.rename(columns=lambda x: x.strip(), inplace=True)

# Bersihkan nilai NaN dan spasi
data_clf['Universitas'] = data_clf['Universitas'].astype(str).str.strip()
data_clf = data_clf[data_clf['Universitas'].notnull()]
data_clf = data_clf[data_clf['Universitas'] != '']
```
[12] ✓ 0.0s Python

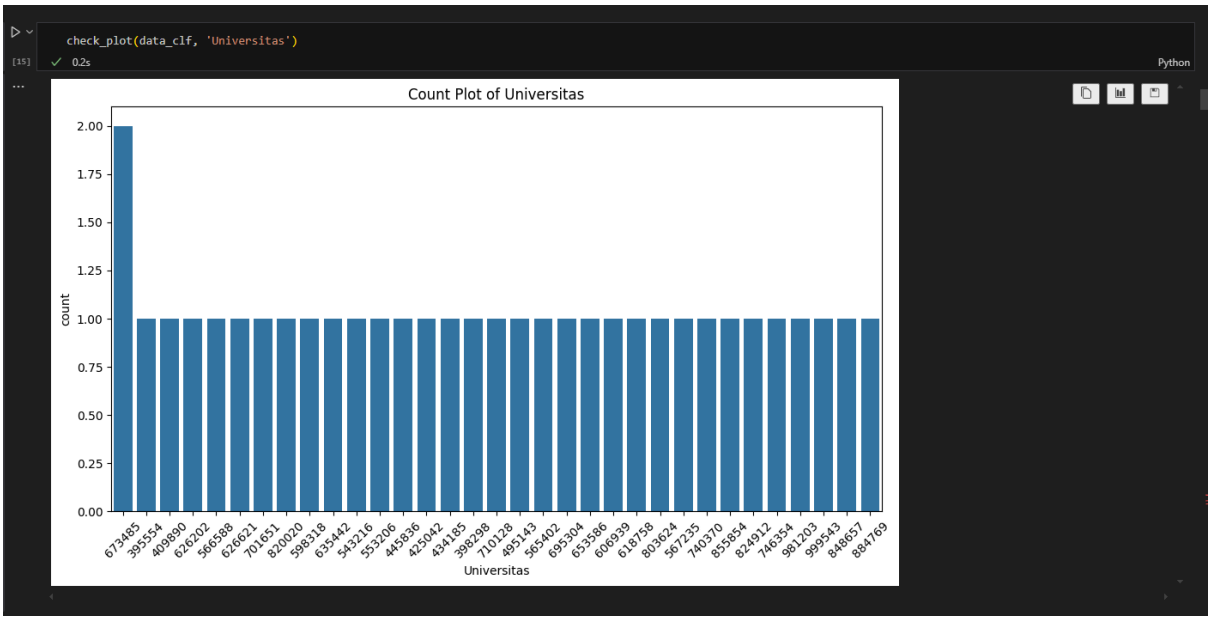
Gambar 4. 6: *Modelling dan evaluasi spasial K-Means clustering 2*

```
def check_plot(data, column_name):
    # Bersihkan kolom yang akan digunakan
    data[column_name] = data[column_name].astype(str).str.strip()
    data = data[data[column_name].notnull()]
    data = data[data[column_name] != '']

    plt.figure(figsize=(10, 6))
    sns.countplot(data=data, x=column_name, order=data[column_name].value_counts().index)
    plt.title(f'Count Plot of {column_name}')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```
[13] ✓ 0.0s Python

Gambar 4.7: *Modelling dan evaluasi spasial K-Means clustering 3*



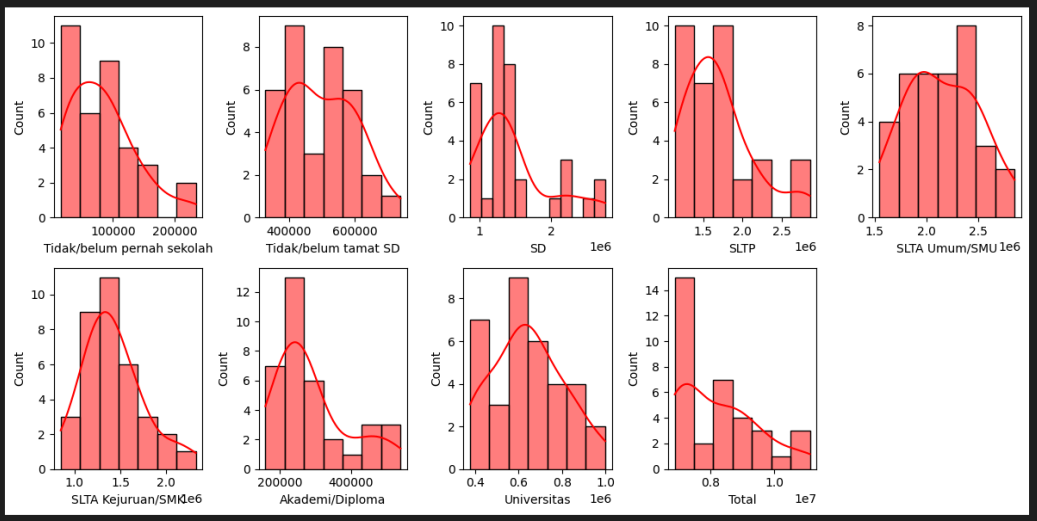
Gambar 4.8: *Modelling dan evaluasi spasial K-Means clustering 4*



Gambar 4.9: *Modelling dan evaluasi spasial K-Means clustering 5*

```python
plt.figure(figsize=(12,6))

# Define the list of numerical features you want to plot
numericals = ['Tidak/belum pernah sekolah', 'Tidak/belum tamat SD', 'SD', 'SLTP', 'SLTA Umum/SMU', 'SLTA Kejuruan/SMK', 'Akademi/Diploma', 'Universitas', 'Total']

features = numericals
for i in range(0, len(features)):
    plt.subplot(2, len(features)//2 + 1, i+1)
    sns.histplot(x=df[features[i]], color='red', kde=True)
    plt.xlabel(features[i])
    plt.tight_layout()
plt.show()
```

[16] ✓ 1.2s Python



Gambar 4.10: *Modelling dan evaluasi spasial K-Means clustering 6*

```python
plt.figure(figsize=(15, 8))

# Box plot untuk Pendidikan Rata-Rata
plt.subplot(1, 2, 1) # 1 baris, 2 kolom, plot ke-1
sns.boxplot(x='Periode', y='Pendidikan_RataRata', data=df, palette='viridis')
plt.title('Distribusi Pendidikan Rata-Rata per Periode', fontsize=16)
plt.xlabel('Periode', fontsize=12)
plt.ylabel('Rata-rata Lama Sekolah (tahun)', fontsize=12)
plt.xticks(rotation=90)
# Anda bisa sesuaikan batas sumbu y jika perlu, contoh: plt.ylim(0, df['Pendidikan_RataRata'].quantile(0.95))

# Box plot untuk Tingkat Pengangguran
plt.subplot(1, 2, 2) # 1 baris, 2 kolom, plot ke-2
sns.boxplot(x='Periode', y='Tingkat_Pengangguran', data=df, palette='viridis')
plt.title('Distribusi Tingkat Pengangguran per Periode', fontsize=16)
plt.xlabel('Periode', fontsize=12)
plt.ylabel('Tingkat Pengangguran (%)', fontsize=12)
plt.xticks(rotation=90)
# Anda bisa sesuaikan batas sumbu y jika perlu, contoh: plt.ylim(0, df['Tingkat_Pengangguran'].quantile(0.95))


plt.tight_layout()
plt.show()
```

[17] ✓ 0.7s Python

Gambar 4.11: *Modelling dan evaluasi spasial K-Means clustering 7*

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Kolom yang berisi informasi periode quarter
kolom_periode_quarter = 'Periode' # Pastikan ini nama kolom yang benar

# Gunakan hasil perhitungan value_counts() dari kolom periode quarter
if kolom_periode_quarter in df.columns:
    periode_counts_df = df[kolom_periode_quarter].value_counts().reset_index()
    periode_counts_df.columns = [kolom_periode_quarter, 'Jumlah Data'] # Sesuaikan nama kolom

    # Mengurutkan periode (opsional tapi disarankan)
    # Ini mungkin memerlukan penyesuaian tergantung format eksak kolom 'Periode' Anda
    # Misalnya, jika formatnya "YYYY Qx", mengurutkan string mungkin sudah cukup
    periode_counts_df = periode_counts_df.sort_values(by=kolom_periode_quarter)

    plt.figure(figsize=(12, 6)) # Sesuaikan ukuran figure jika perlu
    sns.barplot(x=kolom_periode_quarter, y='Jumlah Data', data=periode_counts_df, palette='viridis')

    # Tambahkan label jumlah di atas bar
    for i, count in enumerate(periode_counts_df['Jumlah Data']):
        plt.text(i, count + (0.01 * periode_counts_df['Jumlah Data'].max()), f"{count:,.0f}",
                 ha='center', fontsize=10, color='black')

    plt.title(f'Jumlah Data Per {kolom_periode_quarter}', fontsize=16) # Judul plot
    plt.xlabel(kolom_periode_quarter, fontsize=12) # Label sumbu X
    plt.ylabel('Jumlah Data', fontsize=12) # Label sumbu Y
    plt.xticks(rotation=45) # Rotasi label sumbu X

    plt.tight_layout()
    plt.show()
else:
    print(f"Kolom '{kolom_periode_quarter}' tidak ditemukan di dataframe.")
```

[21] ✓ 0.2s Python

Gambar 4.12: *Modelling dan evaluasi spasial K-Means clustering 8*

```python
# Kelompokkan data berdasarkan 'Periode' dan hitung rata-rata kolom yang relevan
periode_summary = df.groupby('Periode').agg(
    rata_rata_pendidikan=('Pendidikan_RataRata', 'mean'),
    rata_rata_pengangguran=('Tingkat_Pengangguran', 'mean'),
).reset_index()

# Cetak ringkasan per periode
print(periode_summary)
```

[26] ✓ 0.0s Python

Gambar 4.13: *Modelling dan evaluasi spasial K-Means clustering 9*

```python
# Kelompokkan data berdasarkan 'Periode' dan hitung total 'profit' serta jumlah data per periode
periode_summary = df.groupby('Periode').agg(
    total_profit=('profit', 'sum'),
    jumlah_data=('Periode', 'count')  # Menghitung jumlah baris per periode
).reset_index()

# Cetak ringkasan per periode
print(periode_summary)
```

[28] ✓ 0.0s Python

Gambar 4.14: *Modelling dan evaluasi spasial K-Means clustering 10*

```python
plt.figure(figsize=(15, 5))

plt.subplot(1, 2, 1)
plt.plot(k_range, inertias, 'bx-')
plt.xlabel('Jumlah Cluster (k)')
plt.ylabel('Inertia')
plt.title('Elbow Method untuk Optimal k')
plt.grid(True)

plt.subplot(1, 2, 2)
plt.plot(k_range, silhouette_scores, 'rx-')
plt.xlabel('Jumlah Cluster (k)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score untuk Optimal k')
plt.grid(True)

plt.tight_layout()
plt.show()

print("\nNilai Silhouette Score untuk setiap k:")
for k, score in zip(k_range, silhouette_scores):
    print(f"k={k}: {score:.4f}")
```

[34] ✓ 0.2s Python

Gambar 4.15: *Modelling dan evaluasi spasial K-Means clustering 11*

```python
plt.figure(figsize=(12, 6))

scatter = plt.scatter(df['Pendidikan_RataRata'], df['Tingkat_Pengangguran'],
                      c=df['cluster'],
                      cmap='viridis',
                      alpha=0.6)
plt.xlabel('Rata-rata Lama Sekolah (tahun)')
plt.ylabel('Tingkat Pengangguran (%)')
plt.title('Hasil Clustering Provinsi Berdasarkan Pendidikan dan Pengangguran')

plt.colorbar(scatter, label='Cluster')

plt.grid(True)

plt.show()
```

[38] ✓ 0.3s Python

Gambar 4.16: *Modelling dan evaluasi spasial K-Means clustering 12*

Modelling dan evaluasi spasial Decision Tree Regression

Decision Tree Regression

```python
X = df[['Pendidikan_RataRata']]
y = df['Tingkat_Pengangguran']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

tree = DecisionTreeRegressor(max_depth=4, random_state=42)
tree.fit(X_train, y_train)

y_pred = tree.predict(X_test)
r2 = r2_score(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred)

print(f"R2 Score (Decision Tree): {r2:.4f}")
print(f"RMSE (Decision Tree): {rmse:.2f}")
```

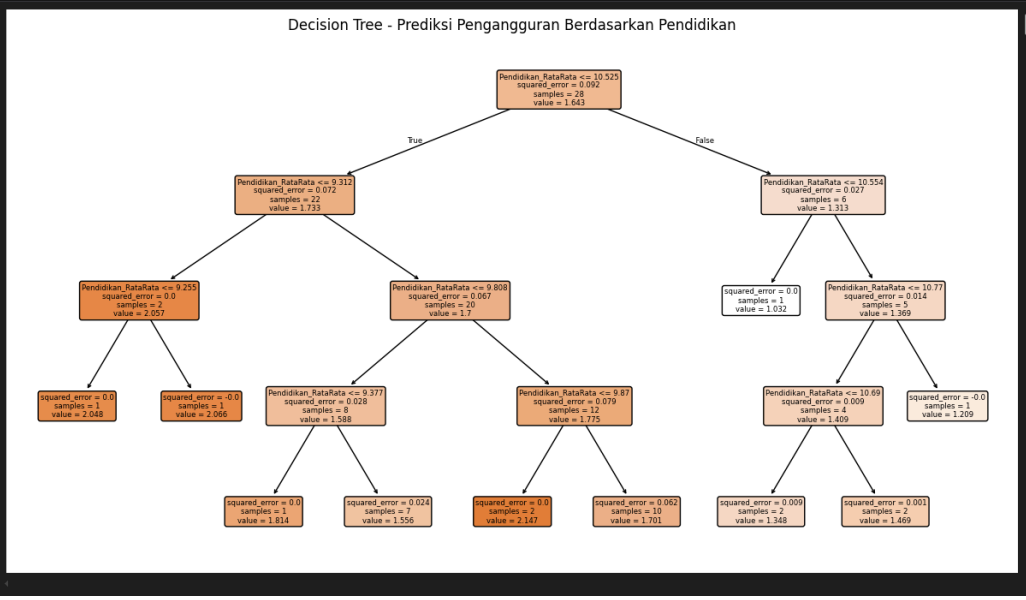
[39] ✓ 0.0s Python

```
R2 Score (Decision Tree): -1.2209
RMSE (Decision Tree): 0.06
```

Gambar 4.17: *Modelling dan evaluasi spasial Decision Tree Regression 1*

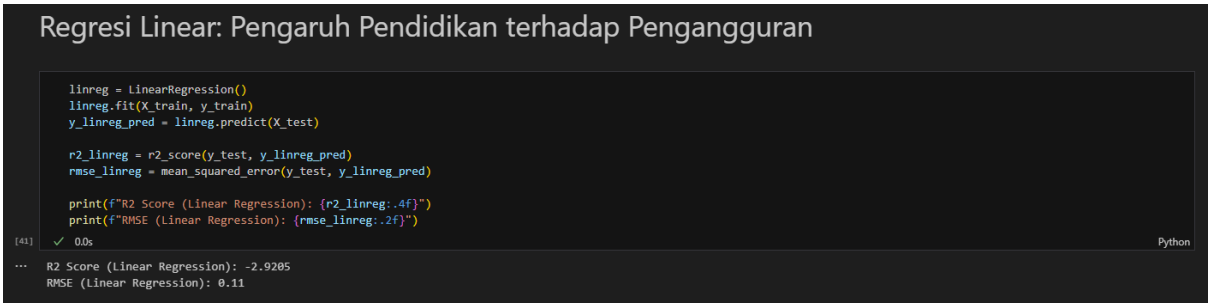
```python
plt.figure(figsize=(15,8))
plot_tree(tree, feature_names=['Pendidikan_RataRata'], filled=True, rounded=True)
plt.title("Decision Tree - Prediksi Pengangguran Berdasarkan Pendidikan")
plt.show()
```

[40] ✓ 0.4s Python

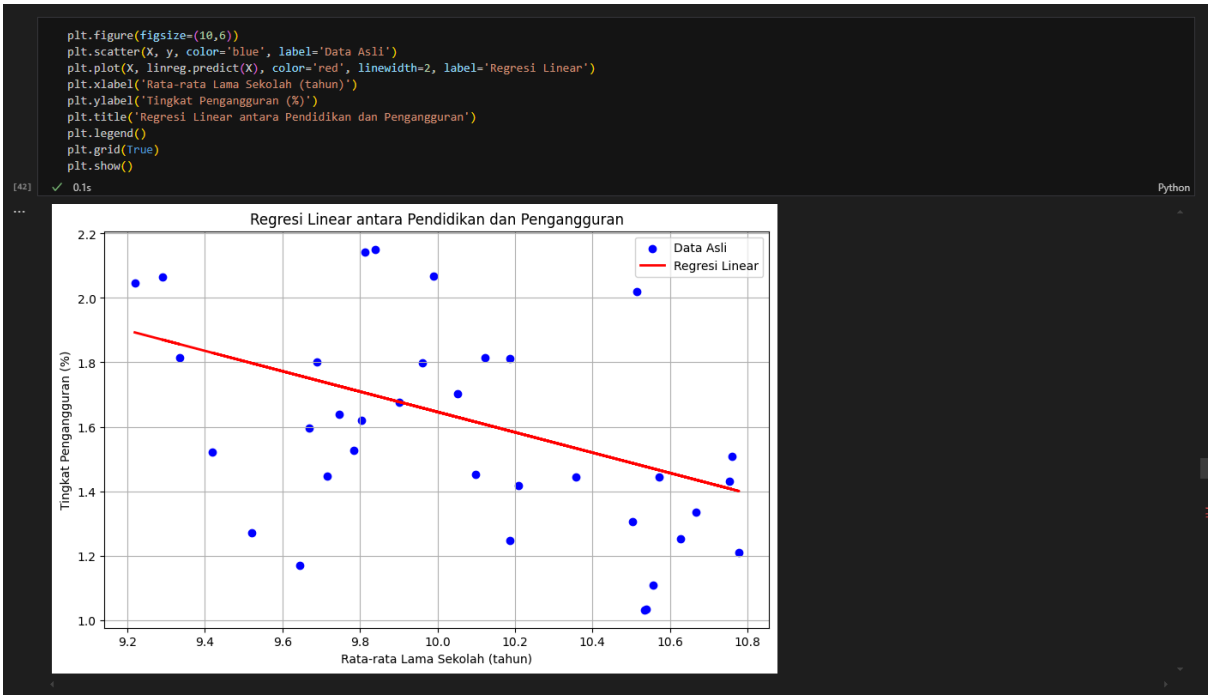


Gambar 4.18: *Modelling dan evaluasi spasial Decision Tree Regression 2*

Modelling dan evaluasi spasial Regresi Linier



Gambar 4.19: *Modelling dan evaluasi spasial Regresi Linier 1*



Gambar 4.20: *Modelling dan evaluasi spasial Regresi Linier 2*

```
import pickle
import os
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler


filename_linreg = 'linear_regression_model.pkl'
try:
    with open(filename_linreg, 'wb') as f:
        pickle.dump(linreg, f)
    print(f"Linear Regression model successfully saved to '{filename_linreg}'")
except NameError:
    print(f"Error: 'linreg' variable not found. Please train your Linear Regression model before saving.")
except Exception as e:
    print(f"Error saving Linear Regression model: {e}")


# Save Decision Tree model
filename_tree = 'decision_tree_model.pkl'
try:
    with open(filename_tree, 'wb') as f:
        pickle.dump(tree, f)
    print(f"Decision Tree model successfully saved to '{filename_tree}'")
except NameError:
     print(f"Error: 'tree' variable not found. Please train your Decision Tree model before saving.")
except Exception as e:
    print(f"Error saving Decision Tree model: {e}")

# Save KMeans Clustering model
filename_kmeans = 'kmeans_model.pkl'
try:
    with open(filename_kmeans, 'wb') as f:
        pickle.dump(final_kmeans, f) # Make sure 'final_kmeans' is the correct variable name
    print(f"KMeans Clustering model successfully saved to '{filename_kmeans}'")
except NameError:
     print(f"Error: 'final_kmeans' variable not found. Please train your KMeans model before saving.")
except Exception as e:
    print(f"Error saving KMeans Clustering model: {e}")

# Save the Scaler used for KMeans clustering
filename_scaler = 'scaler.pkl'
try:
    with open(filename_scaler, 'wb') as f:
[43]
```

Gambar 4.21: *Modelling dan evaluasi spasial Regresi Linier 3*

```
# Save Decision Tree model
filename_tree = 'decision_tree_model.pkl'
try:
    with open(filename_tree, 'wb') as f:
        pickle.dump(tree, f)
    print(f"Decision Tree model successfully saved to '{filename_tree}'")
except NameError:
     print(f"Error: 'tree' variable not found. Please train your Decision Tree model before saving.")
except Exception as e:
    print(f"Error saving Decision Tree model: {e}")

# Save KMeans Clustering model
filename_kmeans = 'kmeans_model.pkl'
try:
    with open(filename_kmeans, 'wb') as f:
        pickle.dump(final_kmeans, f) # Make sure 'final_kmeans' is the correct variable name
    print(f"KMeans Clustering model successfully saved to '{filename_kmeans}'")
except NameError:
     print(f"Error: 'final_kmeans' variable not found. Please train your KMeans model before saving.")
except Exception as e:
    print(f"Error saving KMeans Clustering model: {e}")

# Save the Scaler used for KMeans clustering
filename_scaler = 'scaler.pkl'
try:
    with open(filename_scaler, 'wb') as f:
        pickle.dump(scaler, f) # Make sure 'scaler' is the correct variable name
    print(f"Scaler successfully saved to '{filename_scaler}'")
except NameError:
     print(f"Error: 'scaler' variable not found. Please fit and save your StandardScaler.")
except Exception as e:
    print(f"Error saving Scaler: {e}")
[43]  ✓ 0.0s                                                                    Python

...  Linear Regression model successfully saved to 'linear_regression_model.pkl'
Decision Tree model successfully saved to 'decision_tree_model.pkl'
KMeans Clustering model successfully saved to 'kmeans_model.pkl'
Scaler successfully saved to 'scaler.pkl'
```

Gambar 4.22: *Modelling dan evaluasi spasial Regresi Linier 4*

Deployment Dashboard

```python
import streamlit as st
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor, plot_tree, export_text # Import plot_tree and export_text
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score, r2_score, mean_squared_error
import os

st.set_page_config(
    page_title="Analisis Pengaruh Pendidikan terhadap Tingkat Pengangguran",
    page_icon="📊",
    layout="wide"
)

st.title("Analisis Pengaruh Tingkat Pendidikan terhadap Tingkat Pengangguran di Indonesia")

# --- Data Loading and Preparation ---
st.subheader("Data Loading and Preparation")

uploaded_file = st.file_uploader("Upload dataset CSV", type="csv")

df = None # Initialize df to None

if uploaded_file:
    try:
        df = pd.read_csv(uploaded_file)

        # Data Preparation Steps from your notebook
        tahun_pendidikan = {
            'Tidak/belum pernah sekolah': 0,
            'Tidak/belum tamat SD': 3,
            'SD': 6,
            'SLTP': 9,
            'SLTA Umum/SMU': 12,
            'SLTA Kejuruan/SMK': 12,
            'Akademi/Diploma': 14,
            'Universitas': 16
        }

        # Ensure necessary columns exist before accessing them
```

Gambar 4.23: *Deployment Diagram 1*

```python
        # Ensure necessary columns exist before accessing them
        required_edu_cols = list(tahun_pendidikan.keys()) + ['Total']
        if not all(col in df.columns for col in required_edu_cols):
            st.error("Error: Dataset is missing required education level columns or 'Total'.")
            df = None # Invalidate df if columns are missing
        else:
            df['Total_Pendidikan_Terpenuhi'] = 0
            for level, years in tahun_pendidikan.items():
                df['Total_Pendidikan_Terpenuhi'] += df[level] * years

            # Handle potential division by zero if 'Total' is zero
            df['Pendidikan_RataRata'] = df.apply(
                lambda row: row['Total_Pendidikan_Terpenuhi'] / row['Total'] if row['Total'] > 0 else 0,
                axis=1
            )

            # Simulate 'Tingkat_Pengangguran' as done in your notebook
            # NOTE: In a real scenario, this column would be in your dataset.
            # Since you simulated it, we'll keep the simulation here for demonstration.
            np.random.seed(42)
            df['Tingkat_Pengangguran'] = 12 * np.exp(-0.2 * df['Pendidikan_RataRata']) + np.random.normal(0, 0.3, size=len(df))
            df['Tingkat_Pengangguran'] = df['Tingkat_Pengangguran'].clip(lower=1) # minimal 1%


            st.success("Data loaded and prepared successfully!")
            if 'Periode' in df.columns:
                st.dataframe(df[['Periode', 'Pendidikan_RataRata', 'Tingkat_Pengangguran']].head())
            else:
                st.dataframe(df[['Pendidikan_RataRata', 'Tingkat_Pengangguran']].head())


    except Exception as e:
        st.error(f"Error loading or preparing data: {e}")
        df = None # Invalidate df on any loading/preparation error

# --- Model Loading ---
st.subheader("Model Loading")

# Define the expected model filenames
linear_regression_model_file = 'linear_regression_model.pkl'
decision_tree_model_file = 'decision_tree_model.pkl'
kmeans_model_file = 'kmeans_model.pkl'
scaler_file = 'scaler.pkl' # Assuming you saved the scaler too

models = {}
scaler = None
```

Gambar 4.24: *Deployment Diagram 2*

```python
# Define the expected model filenames
linear_regression_model_file = 'linear_regression_model.pkl'
decision_tree_model_file = 'decision_tree_model.pkl'
kmeans_model_file = 'kmeans_model.pkl'
scaler_file = 'scaler.pkl' # Assuming you saved the scaler too

models = {}
scaler = None
model_loading_success = True

# Check if model files exist before attempting to load
if os.path.exists(linear_regression_model_file) and \
   os.path.exists(decision_tree_model_file) and \
   os.path.exists(kmeans_model_file) and \
   os.path.exists(scaler_file):

    try:
        # Load Linear Regression model
        with open(linear_regression_model_file, 'rb') as f:
            models['Linear Regression'] = pickle.load(f)
        st.success(f"Linear Regression model loaded from '{linear_regression_model_file}'")

        # Load Decision Tree model
        with open(decision_tree_model_file, 'rb') as f:
            models['Decision Tree'] = pickle.load(f)
        st.success(f"Decision Tree model loaded from '{decision_tree_model_file}'")

        # Load KMeans model
        with open(kmeans_model_file, 'rb') as f:
            models['KMeans Clustering'] = pickle.load(f)
        st.success(f"KMeans Clustering model loaded from '{kmeans_model_file}'")

        # Load Scaler
        with open(scaler_file, 'rb') as f:
            scaler = pickle.load(f)
        st.success(f"Scaler loaded from '{scaler_file}'")

    except Exception as e:
        st.error(f"Error loading models: {e}")
        models = {} # Clear models if loading fails
        scaler = None
        model_loading_success = False
else:
    st.warning("Model files not found. Please ensure 'linear_regression_model.pkl', 'decision_tree_model.pkl', 'kmeans_model.pkl', and 'scaler.pkl' are in the same directo
    model_loading_success = False
```

Gambar 4.25: *Deployment Diagram 3*

```python
# --- Model Application and Visualization ---
# Proceed only if data is loaded and models are loaded successfully
if df is not None and models and scaler is not None and model_loading_success:
    st.subheader("Model Results and Visualizations")

    # --- K-Means Clustering with Button ---
    if 'KMeans Clustering' in models:
        st.write("#### K-Means Clustering")

        # Allow user to select the number of clusters
        n_clusters_input = st.slider(
            "Select the number of clusters for K-Means:",
            min_value=2,
            max_value=10,
            value=models['KMeans Clustering'].n_clusters, # Default to loaded model's clusters
            step=1
        )

        # Add a button to trigger clustering
        if st.button(f'Perform K-Means Clustering with {n_clusters_input} clusters'):
            try:
                features_for_clustering = ['Pendidikan_RataRata', 'Tingkat_Pengangguran']
                if not all(col in df.columns for col in features_for_clustering):
                    st.error(f"Clustering features {features_for_clustering} not found in data.")
                else:
                    X_clustering = df[features_for_clustering]
                    X_clustering_scaled = scaler.transform(X_clustering) # Use the loaded scaler

                    # Perform K-Means clustering with the selected number of clusters
                    kmeans_dynamic = KMeans(n_clusters=n_clusters_input, random_state=42, n_init=10) # Specify n_init
                    df['cluster'] = kmeans_dynamic.fit_predict(X_clustering_scaled)

                    cluster_summary = df.groupby('cluster').agg({
                        'Pendidikan_RataRata': 'mean',
                        'Tingkat_Pengangguran': 'mean',
                        'Periode': 'count' if 'Periode' in df.columns else ('Pendidikan_RataRata', 'count') # Count using any column if Periode is missing
                    }).round(2)
                    cluster_summary.rename(columns={cluster_summary.columns[-1]: 'Jumlah Data'}, inplace=True) # Rename the last column to Jumlah Data


                    st.write(f"Cluster Summary ({n_clusters_input} clusters):")
                    st.dataframe(cluster_summary)

                    fig, ax = plt.subplots(figsize=(12, 6))
                    scatter = ax.scatter(df['Pendidikan_RataRata'], df['Tingkat_Pengangguran'],
```

Gambar 4.26: *Deployment Diagram 4*

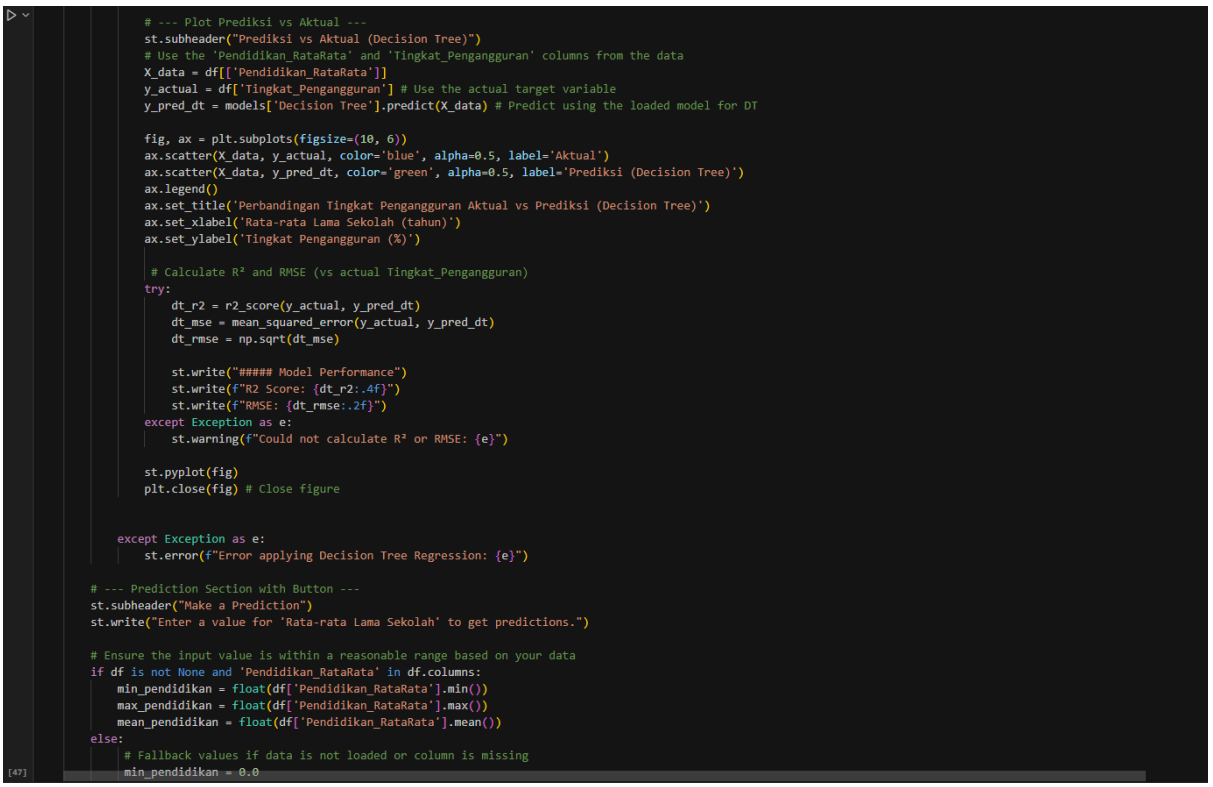
```python
# --- Linear Regression ---
if 'Linear Regression' in models:
    st.write("#### Regresi Linear")
    try:
        # Use the full data for visualization as in your notebook
        X_linreg = df[['Pendidikan_RataRata']]
        y_linreg = df['Tingkat_Pengangguran']

        # Predict using the loaded model
        y_linreg_pred = models['Linear Regression'].predict(X_linreg)

        st.write("Regresi Linear Model Applied.")

        # Plot
        fig, ax = plt.subplots(figsize=(10, 6))
        ax.scatter(X_linreg, y_linreg, color='blue', alpha=0.6, label='Data Asli')
        ax.plot(X_linreg, y_linreg_pred, color='red', linewidth=2, label='Regresi Linear')
        ax.set_xlabel('Rata-rata Lama Sekolah (tahun)')
        ax.set_ylabel('Tingkat Pengangguran (%)')
        ax.set_title('Regresi Linear antara Pendidikan dan Pengangguran')
        ax.legend()
        ax.grid(True)
        st.pyplot(fig)
        plt.close(fig) # Close figure

        # Calculate and display R2 and RMSE
        linreg_r2 = r2_score(y_linreg, y_linreg_pred)
        linreg_mse = mean_squared_error(y_linreg, y_linreg_pred)
        linreg_rmse = np.sqrt(linreg_mse)

        st.write("##### Model Performance")
        st.write(f"R2 Score: {linreg_r2:.4f}")
        st.write(f"RMSE: {linreg_rmse:.2f}")


    except Exception as e:
        st.error(f"Error applying Linear Regression: {e}")

# --- Decision Tree Regression ---
if 'Decision Tree' in models:
    st.write("#### Decision Tree Regression")
    try:
        # Use the full data for visualization/application
        X_tree = df[['Pendidikan_RataRata']]
        y_tree = df['Tingkat_Pengangguran']
```

[47]

Gambar 4.27: *Deployment Diagram 5*

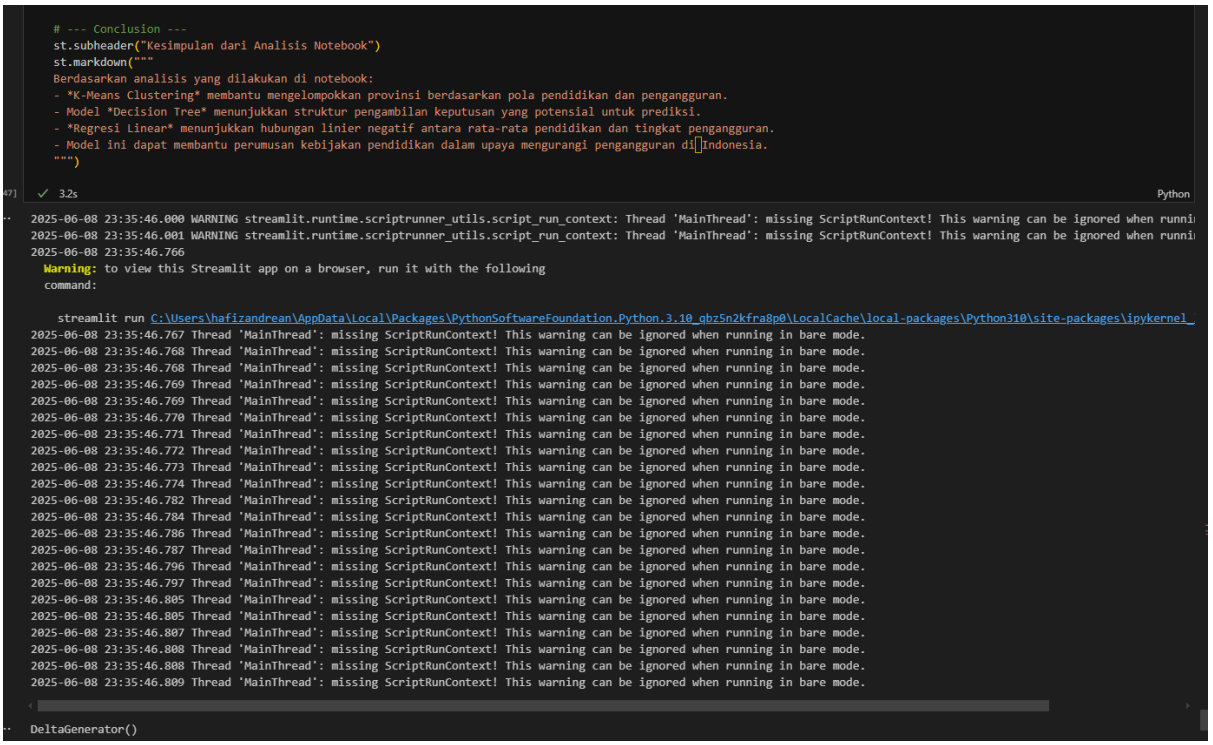
```python
# --- Decision Tree Regression ---
if 'Decision Tree' in models:
    st.write("#### Decision Tree Regression")
    try:
        # Use the full data for visualization/application
        X_tree = df[['Pendidikan_RataRata']]
        y_tree = df['Tingkat_Pengangguran']

        # Predict using the loaded model
        y_tree_pred = models['Decision Tree'].predict(X_tree)

        st.write("Decision Tree Model Applied.")

        # --- Visualisasi Pohon Keputusan ---
        st.subheader("Visualisasi Pohon Keputusan")
        try:
            # Decision Tree plot can be large, handle figure creation
            fig, ax = plt.subplots(figsize=(20, 10))
            # Make sure 'Pendidikan_RataRata' is the correct feature name
            plot_tree(models['Decision Tree'], feature_names=['Pendidikan_RataRata'],
                      filled=True, rounded=True, fontsize=10, ax=ax)
            ax.set_title('Visualisasi Pohon Keputusan') # Set title using ax
            st.pyplot(fig)
            plt.close(fig) # Close figure to free memory
        except Exception as e:
            st.warning(f"Could not generate Decision Tree visualization: {e}")

        # --- Text representation of the tree ---
        st.write("##### Decision Tree Structure (Text)")
        try:
            r = export_text(models['Decision Tree'], feature_names=['Pendidikan_RataRata'])
            st.text(r)
        except Exception as e:
            st.warning(f"Could not generate text tree representation: {e}")
```

Gambar 4.28: *Deployment Diagram 6*



```
            # --- Plot Prediksi vs Aktual ---
            st.subheader("Prediksi vs Aktual (Decision Tree)")
            # Use the 'Pendidikan_RataRata' and 'Tingkat_Pengangguran' columns from the data
            X_data = df[['Pendidikan_RataRata']]
            y_actual = df['Tingkat_Pengangguran'] # Use the actual target variable
            y_pred_dt = models['Decision Tree'].predict(X_data) # Predict using the loaded model for DT

            fig, ax = plt.subplots(figsize=(10, 6))
            ax.scatter(X_data, y_actual, color='blue', alpha=0.5, label='Aktual')
            ax.scatter(X_data, y_pred_dt, color='green', alpha=0.5, label='Prediksi (Decision Tree)')
            ax.legend()
            ax.set_title('Perbandingan Tingkat Pengangguran Aktual vs Prediksi (Decision Tree)')
            ax.set_xlabel('Rata-rata Lama Sekolah (tahun)')
            ax.set_ylabel('Tingkat Pengangguran (%)')

            # Calculate R² and RMSE (vs actual Tingkat_Pengangguran)
            try:
                dt_r2 = r2_score(y_actual, y_pred_dt)
                dt_mse = mean_squared_error(y_actual, y_pred_dt)
                dt_rmse = np.sqrt(dt_mse)

                st.write("##### Model Performance")
                st.write(f"R2 Score: {dt_r2:.4f}")
                st.write(f"RMSE: {dt_rmse:.2f}")
            except Exception as e:
                st.warning(f"Could not calculate R² or RMSE: {e}")

            st.pyplot(fig)
            plt.close(fig) # Close figure


        except Exception as e:
            st.error(f"Error applying Decision Tree Regression: {e}")

    # --- Prediction Section with Button ---
    st.subheader("Make a Prediction")
    st.write("Enter a value for 'Rata-rata Lama Sekolah' to get predictions.")

    # Ensure the input value is within a reasonable range based on your data
    if df is not None and 'Pendidikan_RataRata' in df.columns:
        min_pendidikan = float(df['Pendidikan_RataRata'].min())
        max_pendidikan = float(df['Pendidikan_RataRata'].max())
        mean_pendidikan = float(df['Pendidikan_RataRata'].mean())
    else:
        # Fallback values if data is not loaded or column is missing
        min_pendidikan = 0.0
[47]
```

Gambar 4.29: *Deployment Diagram 7*



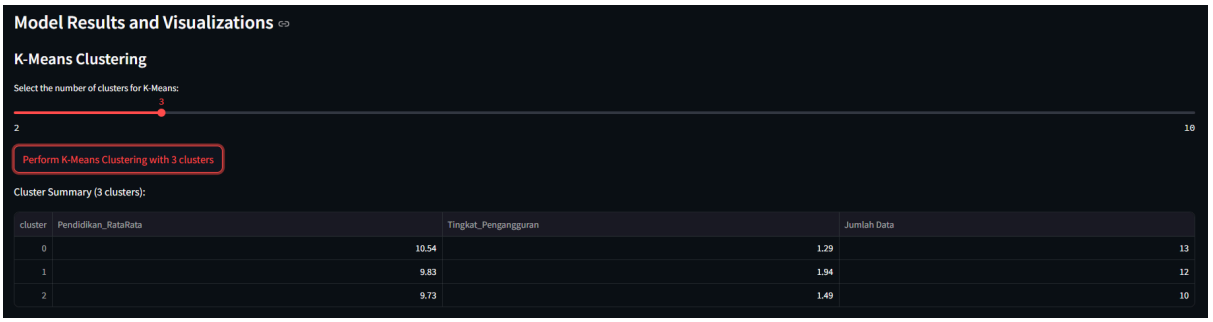
```
    # --- Conclusion ---
    st.subheader("Kesimpulan dari Analisis Notebook")
    st.markdown("""
    Berdasarkan analisis yang dilakukan di notebook:
    - *K-Means Clustering* membantu mengelompokkan provinsi berdasarkan pola pendidikan dan pengangguran.
    - Model *Decision Tree* menunjukkan struktur pengambilan keputusan yang potensial untuk prediksi.
    - *Regresi Linear* menunjukkan hubungan linier negatif antara rata-rata pendidikan dan tingkat pengangguran.
    - Model ini dapat membantu perumusan kebijakan pendidikan dalam upaya mengurangi pengangguran di Indonesia.
    """)
[47] ✓ 3.2s                                                                                    Python

2025-06-08 23:35:46.000 WARNING streamlit.runtime.scriptrunner_utils.script_run_context: Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when runni
2025-06-08 23:35:46.001 WARNING streamlit.runtime.scriptrunner_utils.script_run_context: Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when runni
2025-06-08 23:35:46.766
  Warning: to view this Streamlit app on a browser, run it with the following
  command:

    streamlit run C:\Users\hafizandrean\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages\Python310\site-packages\ipykernel_
2025-06-08 23:35:46.767 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.768 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.768 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.769 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.769 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.770 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.771 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.772 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.773 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.774 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.782 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.784 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.786 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.787 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.796 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.797 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.805 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.805 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.807 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.808 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.808 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-06-08 23:35:46.809 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.

DeltaGenerator()
```

Gambar 4.30: *Deployment Diagram 8*

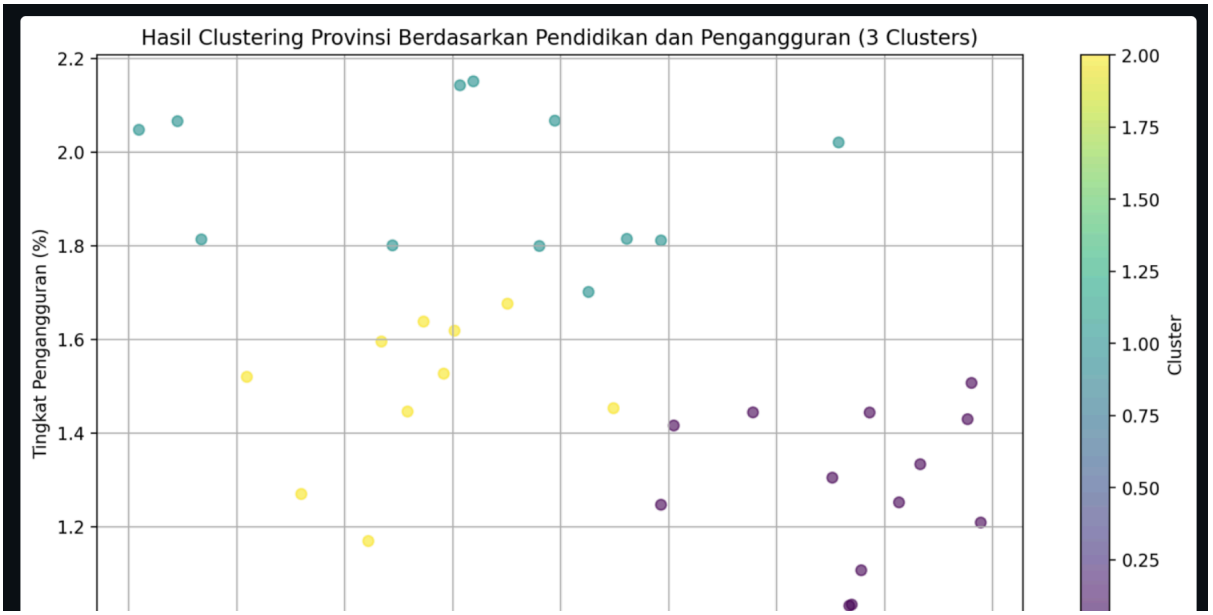
4.1 Hasil

Gambar 4.1.1: *Hasil Interface 1*

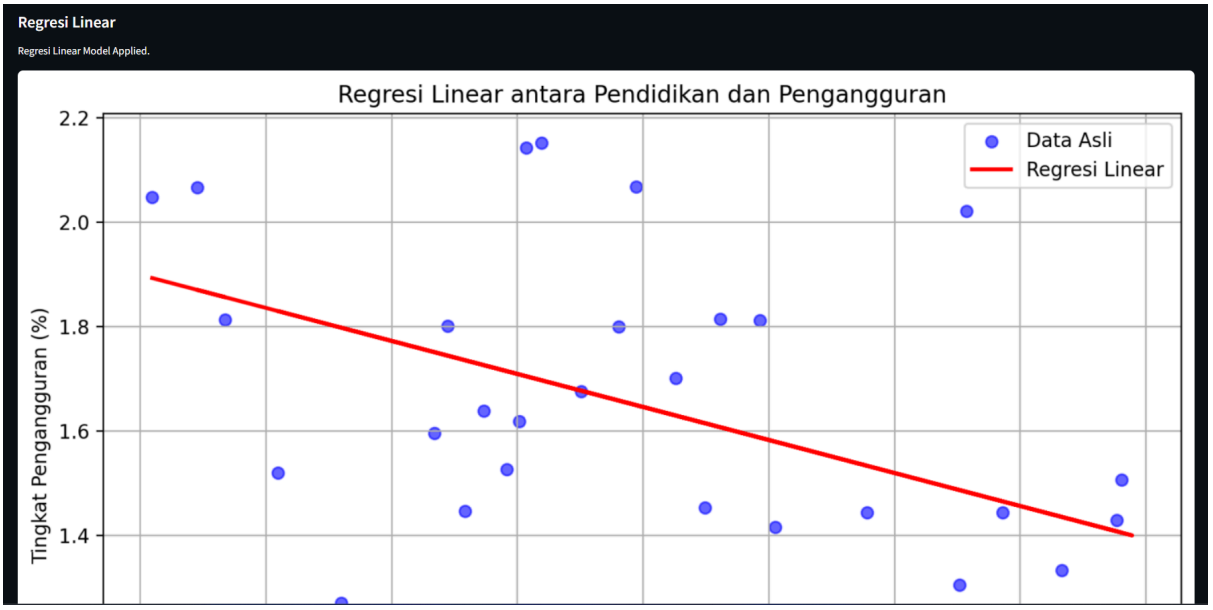
Gambar 4.1.2: *Hasil Interface 2*



Gambar 4.1.3: *Hasil Interface 3*

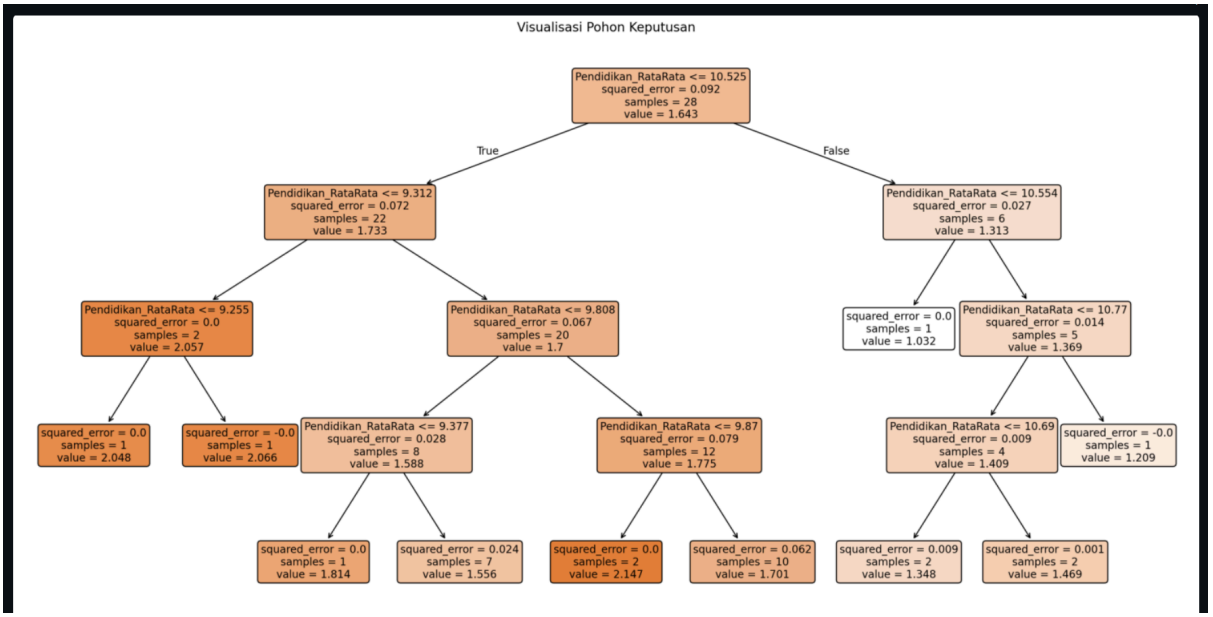


Gambar 4.1.4: *Hasil Interface 4*



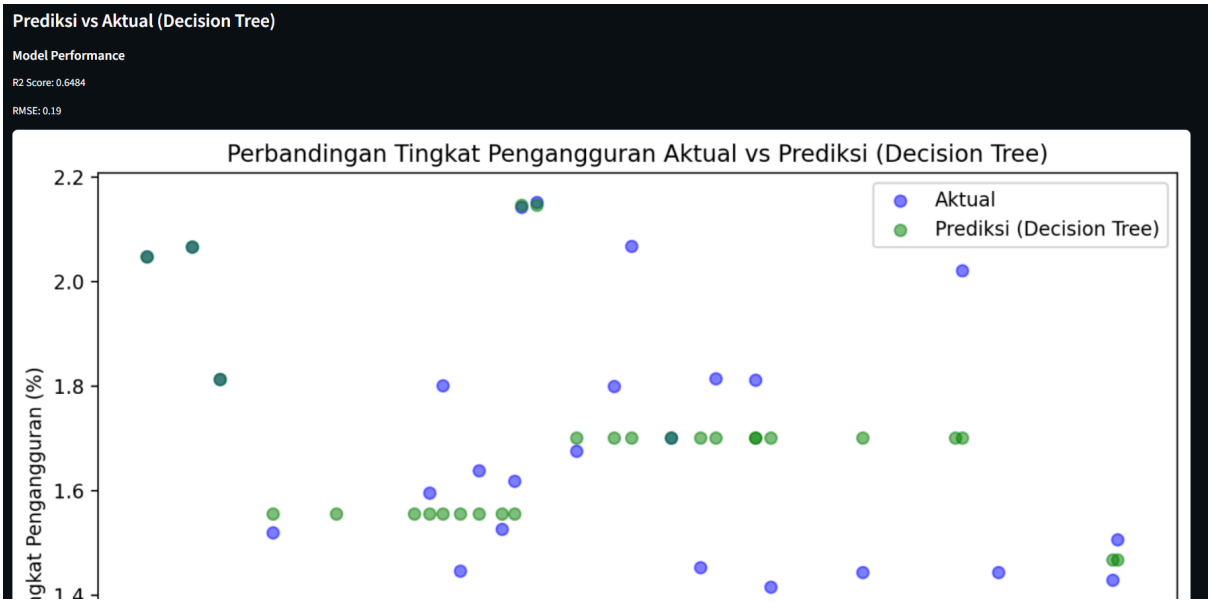
Gambar 4.1.5: *Hasil Interface 5*

Gambar 4.1.6: *Hasil Interface 6*

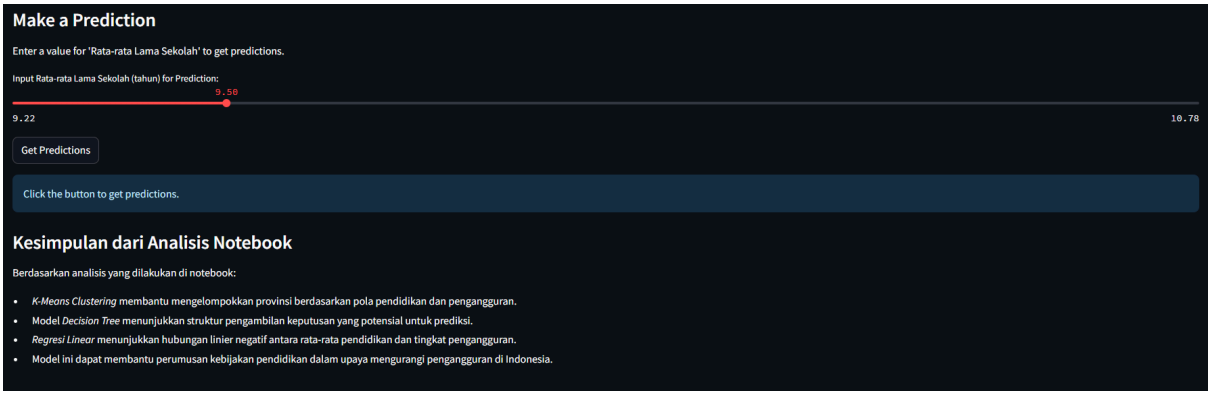


Gambar 4.1.7: *Hasil Interface 7*

Gambar 4.1.8: *Hasil Interface 8*



Gambar 4.1.9: *Hasil Interface 9*



Gambar 4.1.10: *Hasil Interface 10*

4.2 Pembahasan

Hasil dari proses eksplorasi, pemodelan, dan visualisasi dalam penelitian ini memberikan wawasan mendalam mengenai hubungan antara tingkat pendidikan dan tingkat pengangguran di Indonesia. Berdasarkan hasil clustering dengan algoritma K-Means, terlihat bahwa provinsi-provinsi di Indonesia dapat dikelompokkan ke dalam beberapa klaster yang merepresentasikan kesamaan dalam karakteristik pendidikan dan pengangguran. Klaster-klaster tersebut memperlihatkan pola bahwa daerah dengan rata-rata tingkat pendidikan yang rendah cenderung memiliki tingkat pengangguran yang lebih tinggi. Namun, terdapat juga daerah dengan tingkat pendidikan relatif tinggi tetapi tingkat penganggurannya tetap signifikan, khususnya di wilayah urban dan padat penduduk.

Model Decision Tree yang dibangun menunjukkan bahwa jenjang pendidikan merupakan salah satu faktor paling berpengaruh dalam menentukan status pengangguran. Atribut seperti lulusan SMK dan SMA sering muncul sebagai pemisah utama dalam struktur pohon keputusan, yang menandakan bahwa lulusan dari jenjang ini menghadapi risiko pengangguran lebih tinggi dibandingkan dengan lulusan perguruan tinggi atau lulusan

sekolah dasar. Hal ini konsisten dengan fenomena job mismatch di Indonesia, di mana lulusan menengah kejuruan belum sepenuhnya terserap oleh industri.

Hasil dari regresi linier memperkuat temuan sebelumnya dengan menunjukkan korelasi negatif antara rata-rata tingkat pendidikan dan tingkat pengangguran. Semakin tinggi tingkat pendidikan, maka secara umum tingkat pengangguran cenderung menurun. Namun, nilai koefisien determinasi (R^2) yang tidak terlalu tinggi mengindikasikan bahwa pendidikan bukan satu-satunya faktor yang menentukan tingkat pengangguran; faktor lain seperti pertumbuhan ekonomi, kondisi pasar kerja lokal, serta keterampilan non-akademik turut berperan penting.

Jika dibandingkan dengan studi terdahulu seperti yang dilakukan oleh Santoso et al. (2020) dan Wijaya & Lestari (2021), hasil penelitian ini mendukung temuan bahwa model berbasis machine learning seperti Decision Tree dan K-Means sangat efektif untuk memetakan dan menganalisis permasalahan sosial-ekonomi seperti pengangguran. Namun, penelitian ini juga menambahkan kontribusi berupa visualisasi dashboard interaktif yang dapat digunakan untuk eksplorasi data secara dinamis oleh pemangku kebijakan.

Bab V

Penutup

5.1 Kesimpulan

Penelitian ini bertujuan untuk menganalisis pengaruh tingkat pendidikan terhadap tingkat pengangguran di Indonesia dengan pendekatan data mining menggunakan algoritma K-Means Clustering dan Decision Tree, serta dukungan model Regresi Linier. Berdasarkan hasil analisis dan evaluasi, dapat disimpulkan bahwa:

1. Terdapat pola yang cukup kuat antara tingkat pendidikan dan tingkat pengangguran, di mana provinsi dengan rata-rata pendidikan yang rendah cenderung memiliki tingkat pengangguran lebih tinggi.

2. Klaster yang dibentuk oleh K-Means menunjukkan segmentasi wilayah berdasarkan kombinasi karakteristik pendidikan dan pengangguran, yang dapat dimanfaatkan untuk penentuan prioritas kebijakan.

3. Model Decision Tree mampu mengidentifikasi bahwa jenjang pendidikan, khususnya lulusan SMK dan SMA, merupakan indikator signifikan dalam klasifikasi status pengangguran.

4. Regresi linier memperlihatkan bahwa terdapat hubungan negatif antara tingkat pendidikan dan tingkat pengangguran, meskipun tidak sepenuhnya menjelaskan variasi yang terjadi.

5. Visualisasi dashboard interaktif yang dikembangkan memberikan kemudahan dalam pemantauan tren dan penyampaian informasi yang berbasis data.

Dengan demikian, peningkatan tingkat pendidikan harus diiringi dengan peningkatan kualitas kurikulum, pelatihan keterampilan, dan konektivitas dengan dunia kerja agar efektif dalam menurunkan angka pengangguran.

5.2 Saran

Berdasarkan hasil analisis dan kesimpulan yang diperoleh, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut maupun implementasi kebijakan nyata:

1. **Untuk Pemerintah dan Pembuat Kebijakan:** Perlu adanya integrasi yang lebih baik antara sistem pendidikan dan dunia industri. Program pendidikan kejuruan harus disesuaikan dengan kebutuhan pasar tenaga kerja yang terus berkembang, serta didukung oleh pelatihan keterampilan praktis dan kewirausahaan.

2. **Untuk Institusi Pendidikan:** Kurikulum pendidikan perlu dievaluasi secara berkala untuk memastikan bahwa kompetensi lulusan sesuai dengan kebutuhan industri saat

ini. Pendekatan pembelajaran berbasis proyek (project-based learning) dan magang industri dapat menjadi solusi strategis.

3. **Untuk Penelitian Selanjutnya:** Penelitian ini dapat dikembangkan lebih lanjut dengan menggunakan dataset individual-level untuk meningkatkan ketelitian analisis, serta menambahkan variabel lain seperti jenis kelamin, usia, sektor ekonomi, dan teknologi informasi.

4. **Untuk Pengembangan Teknologi Informasi:** Dashboard interaktif dapat dikembangkan lebih lanjut menjadi aplikasi berbasis web yang dapat diakses publik secara luas sebagai alat bantu perumusan kebijakan berbasis data.

Dengan saran-saran tersebut, diharapkan hasil penelitian ini dapat memberikan kontribusi nyata dalam upaya pengurangan pengangguran di Indonesia melalui pendekatan berbasis data dan kolaborasi lintas sektor.

Daftar Pustaka

Badan Pusat Statistik. (2023). *Keadaan Angkatan Kerja di Indonesia Februari 2023*. Jakarta: Badan Pusat Statistik.

Badan Pusat Statistik. (2022). *Statistik Pendidikan 2022*. Jakarta: Badan Pusat Statistik.

Badan Pusat Statistik. (tanpa tahun). *Statistik Tenaga Kerja*. Jakarta: Badan Pusat Statistik.

International Labour Organization. (tanpa tahun). *ILOSTAT Database*. Jenewa: ILO.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., dan Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

Lampiran

Identitas dan Pembagian Tugas Anggota Kelompok

No.	Nama	Nim	Tugas
1.	M. Hafiz Andrean Siregar	102022300045	Mengerjakan laporan, membantu codingan.
2.	Ghifari Derriel Aryasatya	102022330310	Mengerjakan laporan, menyelesaikan codingan.
3.	Muhammad Galih Ilham	102022300364	Mengerjakan laporan, menyelesaikan codingan, mencari dataset.

Tabel Identitas dan Pembagian Tugas Anggota Kelompok