

LAPORAN TUGAS BESAR PEMROGRAMAN PYTHON

TAHUN AJARAN 2025 / 2026

Judul:

**ANALISIS EFISIENSI PENGGUNAAN BAHAN BAKAR PADA MOBIL
MENGUNAKAN MACHINE LEARNING**



Oleh:

IQBAL CHOIRUL RASYID	101012400328
ARIQ BAGASPUTRA SYARANI	101012400116
ZALZABILA S	101012400269
MUHAMMAD RAAFI NURHADI	101012400338

TT-48-03

TELKOM UNIVERSITY BANDUNG

FAKULTAS TEKNIK ELEKTRO

JURUSAN TEKNIK TELEKOMUNIKASI

2025

DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iii
DAFTAR TABEL.....	iv
BAB I PENDAHULUAN.....	5
1.1 LATAR BELAKANG	5
1.2 TUJUAN	5
1.3 RUMUSAN MASALAH.....	6
1.4 PEMBATASAN MASALAH.....	6
1.5 MANFAAT PENELITIAN	7
1.6 SISTEMATIKA PENULISAN.....	7
BAB 2 LANDASAN TEORI.....	8
2.1 DATASET KENDARAAN.....	8
2.2 STATISTIK PADA DATASET	9
2.3 KORELASI PADA DATASET.....	10
2.4 VISUALISASI FITUR DATASET	12
2.5 MACHINE LEARNING	15
2.6 LOGISTIC REGRESSION.....	17
BAB 3 METODOLOGI PENELITIAN	18
3.1 TAHAPAN PENELITIAN	18
3.2 PREPROCESSING DATA.....	19
3.3 PEMBAGIAN DATA.....	21
3.4 MODEL MACHINE LEARNING YANG DIGUNAKAN	21
BAB 4 HASIL DAN PEMBAHASAN	24
4.1 EVALUASI HASIL TRAINING MODEL	24
4.2 PREDIKSI DATA BARU	27
BAB 5 KESIMPULAN DAN SARAN	32
5.1 KESIMPULAN.....	32
5.2 SARAN.....	32
DAFTAR PUSTAKA	xxxiii

DAFTAR GAMBAR

Gambar 2.1 Heatmap Korelasi Antar Fitur.....	11
Gambar 2.2 Hubungan Antara Berat Dengan MPG	13
Gambar 2.3 Hubungan Antara Kapasitas Mesin Dengan MPG	14
Gambar 2.4 Hubungan Antara Kekuatan Mesin Dengan MPG.....	15
Gambar 2.5 Fungsi Sigmoid	17
Gambar 3.1 Diagram Alir Tahapan Penelitian	18
Gambar 4.1 Confusion Matrix	25
Gambar 4.2 Hasil Analisis Toyota Fortuner 2.4 Diesel.....	29
Gambar 4.3 Hasil Analisis Honda HR-V 1.5.....	30
Gambar 4.4 Hasil Analisis Mitsubishi Pajero Sport 3.0	30
Gambar 4.5 Hasil Analisis Toyota Alphard 3.5.....	30
Gambar 4.6 Hasil Analisis Nissan Grand Livina.....	30
Gambar 4.7 Hasil Analisis Toyota Avanza 2015.....	31

DAFTAR TABEL

Tabel 2.1 Statistik Pada Dataset.....	10
---------------------------------------	----

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Mobil merupakan alat transportasi yang berguna untuk mengantarkan penumpang dan barang bawaan dalam jumlah banyak ke tempat yang jauh. Banyak orang memilih jenis transportasi ini karena banyak hal, misalnya seperti agar tidak kehujanan, lebih nyaman saat perjalanan jauh, dan alasan lainnya. Namun semakin banyak pengguna transportasi terutama mobil dapat menyebabkan masalah serius. Jonathan (2025) menyatakan bahwa jumlah kendaraan di dunia telah meningkat secara drastis dan menyentuh angka sekitar 1,644 miliar unit yang melintasi jalan raya global.

Masalah terpenting yang dihasilkan oleh kendaraan berbahan bakar minyak (BBM) adalah polusi udara yang mencemari lingkungan dan penggunaan energi tak terbarukan sebagai bahan bakar. Banyak studi yang meneliti tentang penggunaan bahan bakar alternatif sebagai pengganti BBM seperti contohnya kendaraan listrik dan juga studi tentang pengurangan emisi karbon pada kendaraan BBM yang sudah ada. Namun dari banyak studi ini, sangat sulit untuk merubah kebiasaan orang dalam menggunakan kendaraan selain berbahan bakar minyak dimana beberapa dari mereka enggan mengganti dengan alasan kendaraan sebelumnya masih layak pakai ataupun mahal jika membeli kendaraan baru.

Oleh karena itu, dibutuhkan sebuah metode untuk mengklasifikasikan apakah sebuah kendaraan terutama mobil menjadi irit ataupun boros berdasarkan penggunaan bahan bakarnya.

1.2 TUJUAN

Tujuan dari dibuatnya laporan ini adalah sebagai berikut:

1. Memenuhi persyaratan pembuatan tugas besar mata kuliah Pemrograman Python.
2. Menganalisis pola penggunaan bahan bakar pada kendaraan mobil berdasarkan karakteristiknya.

3. Menciptakan model klasifikasi irit atau borosnya kendaraan menggunakan Machine Learning.
4. Memberikan rekomendasi kendaraan yang lebih efisien dan ramah lingkungan.

1.3 RUMUSAN MASALAH

Rumusan Masalah yang dapat ditemukan pada penelitian ini adalah sebagai berikut:

1. Bagaimana cara mengklasifikasikan kendaraan bermotor, khususnya mobil, ke dalam kategori irit atau boros berdasarkan penggunaan bahan bakar?
2. Fitur atau parameter apa saja yang paling berpengaruh terhadap tingkat konsumsi bahan bakar kendaraan?
3. Bagaimana hubungan (korelasi) antara karakteristik kendaraan dengan konsumsi bahan bakarnya?
4. Seberapa baik model klasifikasi yang dibangun dalam membedakan kendaraan irit dan boros berdasarkan data yang tersedia?
5. Apakah metode klasifikasi yang digunakan dapat memberikan hasil yang cukup akurat untuk membantu analisis efisiensi bahan bakar kendaraan?

1.4 PEMBATASAN MASALAH

Penulisan laporan ini disertai pembatasan masalah agar lebih terfokus dan terstruktur pada tema dan topik yang akan dibahas. Adapun pembatasan masalah pada laporan pemrograman python ini adalah:

1. Objek penelitian dibatasi pada kendaraan berjenis mobil, dan tidak mencakup kendaraan lain seperti sepeda motor, bus, atau truk.
2. Hanya membahas tentang data dan karakteristik mobil yang telah disediakan pada dataset.
3. Klasifikasi kendaraan hanya dibagi ke dalam dua kategori, yaitu kendaraan irit dan boros, berdasarkan tingkat konsumsi bahan bakar.
4. Metode yang digunakan difokuskan pada analisis data dan klasifikasi, tanpa membahas secara rinci aspek mekanis mesin atau desain kendaraan secara fisik.
5. Parameter yang dianalisis dibatasi pada beberapa fitur utama kendaraan yang memiliki korelasi terbesar terhadap konsumsi bahan bakar.

1.5 MANFAAT PENELITIAN

Manfaat dari penelitian yang dilakukan ini adalah sebagai berikut :

1. Membantu pengguna kendaraan dalam mengidentifikasi tingkat efisiensi bahan bakar mobil berdasarkan karakteristik kendaraan yang dimiliki.
2. Menunjukkan bagaimana pemilihan fitur berdasarkan korelasi dapat meningkatkan fokus dan efisiensi dalam pembuatan model klasifikasi.

1.6 SISTEMATIKA PENULISAN

Untuk mempermudah dalam memahami pembahasan laporan pemrograman python ini secara menyeluruh, maka penyajian materi laporan ini disusun secara sistematis yaitu sebagai berikut:

1. BAB 1 PENDAHULUAN

Pada bab ini akan membahas tentang latar belakang, tujuan, rumusan masalah, pembatasan masalah, manfaat penelitian, dan sistematika penulisan.

2. BAB 2 LANDASAN TEORI

Pada bab ini akan membahas tentang dataset kendaraan, statistik deskriptif, korelasi data, machine learning, dan logistic regression.

3. BAB 3 METODOLOGI PENELITIAN

Pada bab ini berisi tentang alur penelitian, preprocessing data, pembagian data, dan model machine learning.

4. BAB 4 HASIL DAN PEMBAHASAN

Pada bab ini akan membahas tentang visualisasi data, hasil training model, dan prediksi data baru.

5. BAB 5 KESIMPULAN DAN SARAN

Pada bab ini akan membahas tentang kesimpulan yang didapatkan, dan saran untuk penelitian kedepannya.

6. DAFTAR PUSTAKA

Daftar pustaka akan berisi dokumentasi, dan sumber dataset.

BAB 2

LANDASAN TEORI

2.1 DATASET KENDARAAN

Dataset kendaraan yang digunakan dalam penelitian ini berisi beberapa atribut yaitu:

- **Name**
Name atau nama merupakan nama dari kendaraan tersebut.
- **Miles_per_Gallon**
Miles per Gallon (MPG) merupakan satuan untuk mengukur jarak yang dapat ditempuh kendaraan ketika menghabiskan bahan bakar sebanyak satu gallon. Semakin tinggi nilai MPG pada sebuah kendaraan, maka semakin irit kendaraan tersebut.
- **Horsepower**
Horsepower atau tenaga kuda merupakan satuan dari tenaga yang dihasilkan oleh mesin kendaraan. Semakin besar horsepower yang dihasilkan kendaraan, maka semakin tinggi performa kendaraan tersebut.
- **Weight_in_lbs**
Weight atau berat merupakan bobot yang dimiliki oleh sebuah kendaraan. Satuan dari berat yang digunakan pada dataset adalah lbs.
- **Cylinders**
Cylinders atau silinder merupakan wadah tempat piston bergerak dan tempat terjadinya pembakaran pada mesin kendaraan. Semakin banyak silinder pada sebuah mesin memungkinkan lebih banyak bahan bakar dan udara untuk masuk ke ruang bakar.
- **Displacement**

Displacement atau kapasitas mesin merupakan volume dari ruang bakar mesin yang diukur dalam satuan liter (L), sentimeter kubik (CC), dan inchi kubik (Cu In). Semakin besar nilai dari kapasitas mesin maka semakin banyak bahan bakar dan udara yang dapat masuk ke ruang bakar.

- **Acceleration**

Acceleration atau percepatan merupakan waktu percepatan dari sebuah kendaraan. Percepatan diukur dari seberapa cepat kendaraan bergerak dari kondisi diam. Akselerasi diukur dalam satuan Miles per hour (Mph) dan Kilometer per hour (Kmh)

- **Year**

Year atau tahun merupakan tahun kapan kendaraan tersebut diproduksi.

- **Origin**

Origin atau asal merupakan lokasi dimana kendaraan tersebut diproduksi.

2.2 STATISTIK PADA DATASET

Statistik pada dataset digunakan untuk memberikan gambaran umum mengenai karakteristik data yang dianalisis. Melalui statistik, dapat diketahui nilai rata-rata, nilai tengah, tingkat penyebaran, serta rentang data dari setiap variabel penelitian. Informasi ini penting sebagai langkah awal untuk memahami pola dasar data sebelum dilakukan analisis lebih lanjut.

Berikut ini merupakan kode yang digunakan untuk menganalisis dataset.

```
import pandas as pd
df = pd.read_json("X:\Code\Tubes_Phyton_Kel4\Dataset\Raw\cars2.json")
df = df.dropna()
print(df.describe())
```

Output dari kode ini adalah sebagai berikut :

	Miles_per_Gallon	Cylinders	Displacement	Horsepower	Weight_in_lbs	Acceleration
N	392	392	392	392	392	392
Mean	23,4459	5,4719	194,4119	104,4693	2977,5841	15,5413
Standar Deviasi	7,805	1,7057	104,644	38,4911	849,4025	2,7588
Min	9	3	68	46	1613	8
Max	46,6	8	455	230	5140	24,8
Q1	17	4	105	75	2225,25	13,775
Q2	22,75	4	151	93,5	2803,5	15,5
Q3	29	8	275,75	126	3614,75	17,025

Tabel 2.1 Statistik Pada Dataset

Berdasarkan tabel 2.1, dapat terlihat persebaran data antara masing masing variabel numerik dimana N merupakan banyaknya data pada masing masing variabel, mean merupakan nilai rata rata, standar deviasi merupakan seberapa tersebar data, min merupakan nilai terkecil pada data, max merupakan nilai terbesar pada data, Q1 merupakan kuartil 1, Q2 merupakan kuartil 2, dan Q3 merupakan kuartil 3. Perhitungan dari masing masing data dilakukan setelah menghapus nilai data yang kosong.

Berdasarkan statistik ini, dapat disimpulkan bahwa :

- Berdasarkan kuartil 1 dan 3 pada MPG, terdapat 25% kendaraan yang memiliki MPG kurang dari 17 dan 75% kendaraan yang memiliki MPG lebih dari 29.
- Standar deviasi dari displacement, horsepower, dan weight memiliki nilai yang besar dan menandakan masing masing variabel memiliki persebaran data yang tidak wajar dan banyak yang menjauhi nilai rata rata.
- Karena nilai kuartil 2 dan rata rata dari MPG hampir sama, maka persebaran antara mobil irit dan boros seimbang sehingga tidak ada nilai ekstrim pada rentang data.
- Rata rata dan kuartil 2 dapat dijadikan nilai batas untuk menentukan apakah kendaraan irit atau boros.

2.3 KORELASI PADA DATASET

Korelasi adalah keterkaitan antara dua data ataupun lebih untuk melihat apakah ketika satu variabel naik, maka variabel lain ikut naik atau malah turun. Pada dataset, korelasi digunakan untuk melihat hubungan variabel MPG dengan variabel lainnya dengan tujuan untuk mencari variabel mana yang akan dijadikan fitur dalam training model prediksi. Korelasi dapat dibuat dengan menggunakan pandas dan menghiraukan variabel yang tidak memiliki numerik didalamnya.

Berikut ini merupakan kode yang digunakan untuk melihat korelasi antar variabel:

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

df = pd.read_json('X:\Code\Tubes_Phyton_Kel4\Dataset\Raw\cars2.json')

df_numeric = df.select_dtypes(include=['int64', 'float64'])

corr_matrix = df_numeric.corr()

plt.figure(figsize=(10, 8))

sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap='coolwarm',

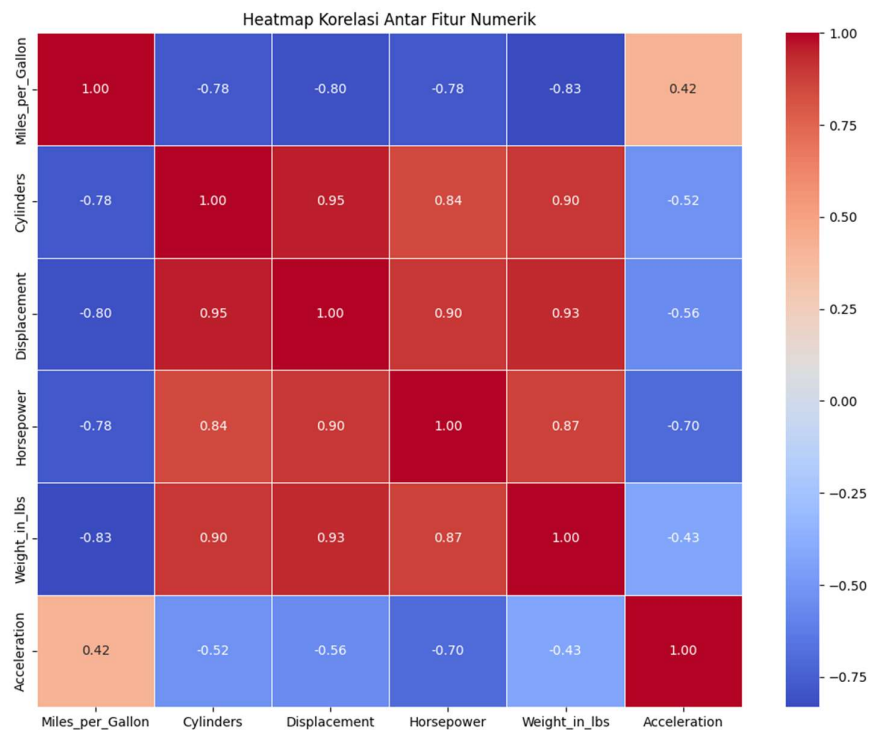
            linewidths=0.5)

plt.title("Heatmap Korelasi Antar Fitur Numerik")

plt.tight_layout()

plt.show()
```

Output dari kode ini adalah sebagai berikut :



Gambar 2.1 Heatmap Korelasi antar Fitur

Dari gambar 2.1, terlihat bahwa Cylinders (-0,78), Displacement (-0,80), Horsepower (-0,78), dan Weight (-0,83) berbanding terbalik dengan nilai MPG (mendekati -1) sedangkan Acceleration (0,42) hampir tidak memiliki korelasi dengan MPG (mendekati 0). Karena berbanding terbalik, jika nilai dari Weight, Displacement, dan Horsepower naik maka nilai dari MPG akan turun dan sebaliknya.

Berdasarkan hal tersebut, dapat disimpulkan bahwa variabel yang dapat digunakan sebagai fitur pada training model prediksi adalah Weight, Displacement, dan Horsepower. Variabel Cylinders tidak digunakan karena Cylinder dan Displacement memiliki keterkaitan yang sama terhadap konsumsi bahan bakar yaitu sama sama memungkinkan lebih banyak bahan bakar dan udara untuk masuk ke ruang bakar.

2.4 VISUALISASI FITUR DATASET

Visualisasi merupakan gambaran dari hubungan data dengan data lainnya dan ditampilkan dalam bentuk grafik maupun media lain. Visualisasi berguna untuk membantu memahami dalam melihat data secara visual dibandingkan dengan angka maupun nilai dari data tersebut.

Berikut ini adalah kode dari hubungan antara berat kendaraan dengan MPG nya :

```
import pandas as pd

import matplotlib.pyplot as plt

df =pd.read_json("X:\Code\Tubes_Phyton_Kel4\Dataset\Raw\cars2.json")

plt.scatter(df['Weight_in_lbs'], df['Miles_per_Gallon'])

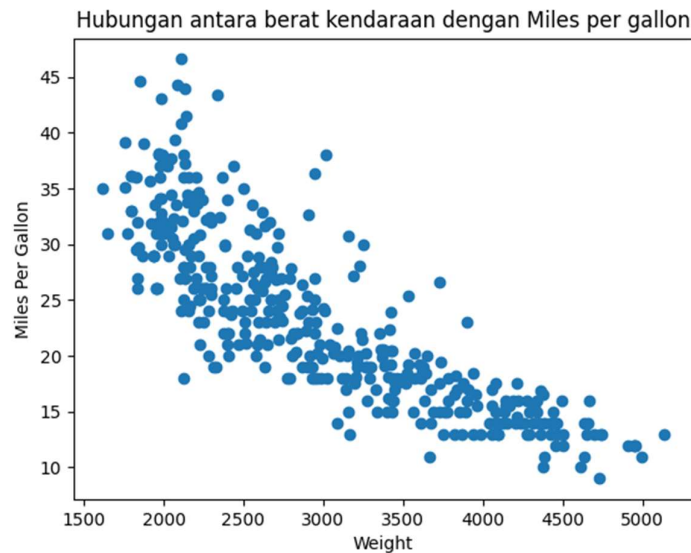
plt.xlabel("Berat")

plt.ylabel("Miles Per Gallon")

plt.title("Hubungan antara berat kendaraan dengan Miles per gallon")

plt.show()
```

Output dari kode ini adalah sebagai berikut :



Gambar 2.2 Hubungan Antara Berat Dengan MPG

Berdasarkan hasil visualisasi gambar 2.2, terlihat bahwa persebaran berat kendaraan terhadap MPG membentuk pola melengkung dimana semakin berat kendaraan maka konsumsi bahan bakarnya akan meningkat dan membuat nilai MPG menurun. Berdasarkan visual ini, dapat disimpulkan bahwa tidak ada data ekstrim yang berarti persebaran irit maupun boros kendaraan berdasarkan perbandingan berat dengan MPG nya terbagi dengan rata.

Berikut ini adalah kode dari hubungan antara kapasitas mesin kendaraan dengan MPG nya :

```
import pandas as pd

import matplotlib.pyplot as plt

df = pd.read_json("X:\Code\Tubes_Phyton_Kel4\Dataset\Raw\cars2.json")

plt.scatter(df['Displacement'], df['Miles_per_Gallon'])

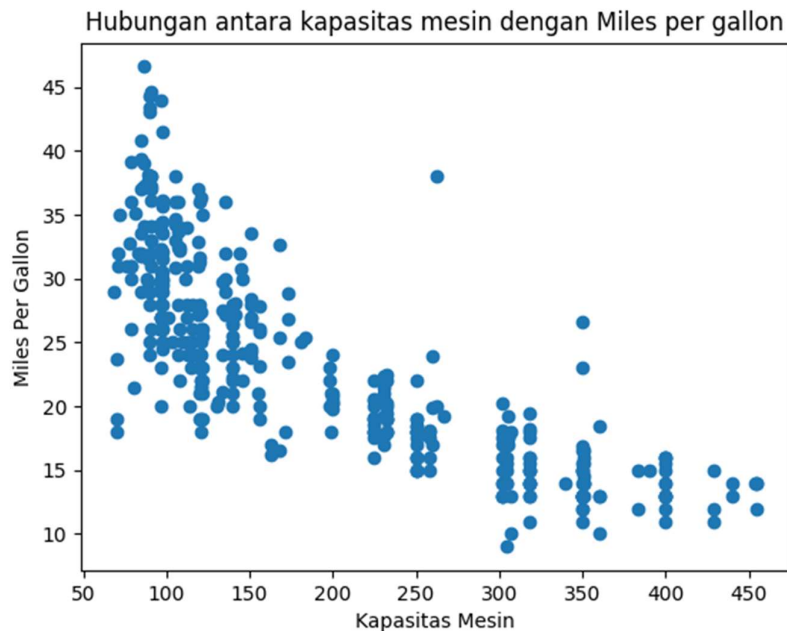
plt.xlabel("Displacement")

plt.ylabel("Miles Per Gallon")

plt.title("Hubungan antara kapasitas mesin dengan Miles per gallon")

plt.show()
```

Output dari kode ini adalah sebagai berikut :



Gambar 2.3 Hubungan Antara Kapasitas Mesin Dengan MPG

Berdasarkan hasil visualisasi gambar 2.3, terlihat bahwa persebaran data dari kapasitas mesin terhadap MPG nya cukup merata, namun ada sedikit data yang ekstrim di sekitar angka 250 pada kapasitas mesin dan 39 pada MPG nya. Hal ini terbukti karena nilai dari standar deviasi kapasitas mesin mencapai nilai 104,644 dimana banyak data kapasitas mesin yang menjauhi nilai rata rata nya.

Berikut ini adalah kode dari hubungan antara tenaga mesin dengan MPG nya :

```
import pandas as pd

import matplotlib.pyplot as plt

df=pd.read_json("X:\Code\Tubes_Phyton_Kel4\Dataset\Raw\cars2.json")

plt.scatter(df[ 'Horsepower' ], df[ 'Miles_per_Gallon' ])

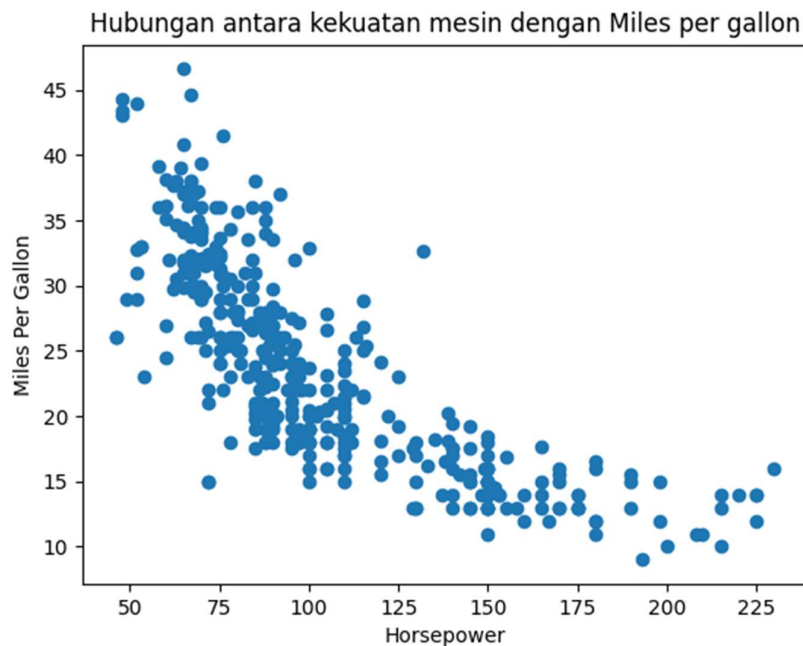
plt.xlabel("Tenaga mesin")

plt.ylabel("Miles Per Gallon")

plt.title("Hubungan antara kekuatan mesin dengan Miles per Gallon")

plt.show()
```

Output dari kode ini adalah sebagai berikut :



Gambar 2.4 Hubungan Antara Kekuatan Mesin Dengan MPG

Berdasarkan hasil visualisasi gambar 2.4, terlihat bahwa hasil persebaran data kekuatan mesin dengan MPG nya tidak cukup merata dikarenakan persebaran lebih banyak di sebelah kiri dibandingkan sebelah kanannya. Selain itu ada beberapa nilai ekstrim yang terlihat di sekitar data 50 sampai 150 kekuatan mesin.

2.5 MACHINE LEARNING

Machine learning (ML) adalah cabang dari Kecerdasan Buatan (AI) yang melatih sistem komputer untuk belajar dari data, mengenali pola, dan membuat keputusan atau prediksi tanpa diprogram secara eksplisit untuk setiap tugasnya sehingga sistem dapat meningkatkan performa seiring waktu dengan pengalaman. ML terbagi menjadi beberapa jenis tergantung dengan bagaimana model ML tersebut dilatih dan algoritma apa yang ia gunakan.

Beberapa jenis pelatihan pelatihan ML yaitu :

1. Supervised Learning

Supervised learning adalah cara melatih model ML menggunakan data latih yang memiliki label atau kunci jawaban yang diketahui.

2. Unsupervised Learning

Unsupervised learning adalah cara melatih model ML ketika data tidak memiliki label. Konsep belajar ini menggunakan kesamaan karakteristik dari masing-masing data alih-alih menggunakan label sebagai kunci jawaban.

3. Reinforcement Learning

Reinforcement learning adalah cara melatih model ML menggunakan sistem reward dan penalti. Konsep ini membuat model dapat berkembang menjadi lebih cerdas setiap kali ia belajar.

Penelitian ini menggunakan jenis pelatihan supervised learning karena menggunakan data latih dan kunci jawaban atau label dalam melatih model. Pada jenis ini, terdapat beberapa dua kategori prediksi yaitu klasifikasi dan regresi dimana klasifikasi berarti model dilatih supaya dapat mengenal perbedaan, mengklasifikasikan data, maupun mengelompokkan data. Sedangkan regresi berarti model dilatih untuk melakukan prediksi nilai nyata riil maupun kontinu. Algoritma model dapat dipilih berdasarkan dua kategori ini.

- **Regresi**

1. Regresi Linear
2. Regresi Polinomial
3. Regresi Ridge & Lasso
4. Dan lainnya.

- **Klasifikasi**

1. KNN
2. SVM
3. Regresi Logistik
4. Decision Tree
5. Random Forest
6. Dan Lainnya

2.6 LOGISTIC REGRESSION

Logistic Regression adalah algoritma yang mengklasifikasikan sebuah data menjadi biner (0/1), iya/tidak, irit/boros dan lainnya. Algoritma ini menggunakan fungsi sigmoid yang memetakan hasil regresi ke bentuk biner.

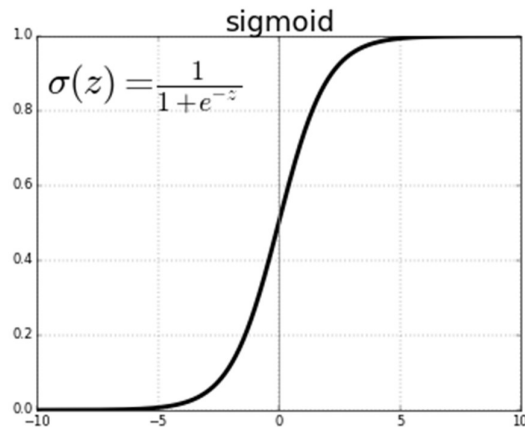
Fungsi sigmoid memiliki bentuk sebagai berikut :

$$\sigma(Z) = \frac{1}{1 + e^{-Z}}$$

Dimana :

$$Z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_N X_N$$

Grafik dari fungsi sigmoid adalah sebagai berikut :



Gambar 2.5 Fungsi Sigmoid

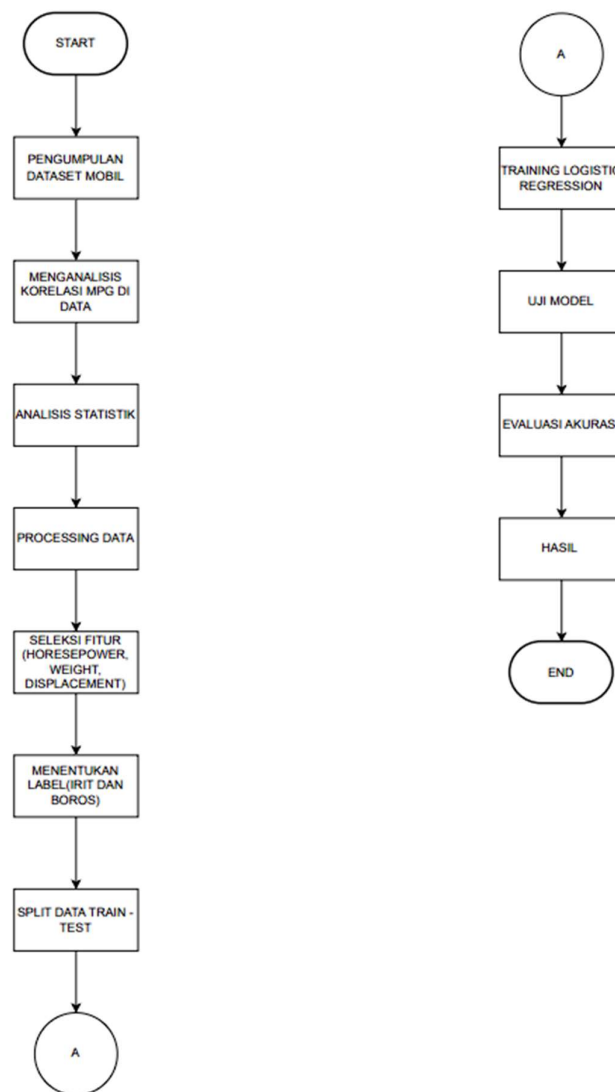
Mencari Z dari regresi logistik tidak sama dengan mencari Y pada regresi linear dikarenakan koefisien dari regresi logistik memerlukan perhitungan berulang sampai koefisien tersebut konvergen.

BAB 3

METODOLOGI PENELITIAN

3.1 TAHAPAN PENELITIAN

Metodologi penelitian pada laporan ini disusun secara sistematis agar proses analisis dan pembuatan model dapat dilakukan secara terarah. Berikut merupakan diagram alir dari tahapan penelitian :



Gambar 3.1 Diagram Alir Tahapan Penelitian

Tahap pertama dari penelitian ini adalah mengumpulkan data. Peneliti mengambil dataset mobil amerika untuk digunakan sebagai sumber utama pada analisis ini. Berikutnya adalah menganalisis korelasi antara masing masing variabel terhadap konsumsi bahan bakarnya. Tahap berikutnya adalah melakukan analisis statistik pada keseluruhan data untuk mengetahui rata rata, kuartil, standar deviasi, dan nilai minimum dan maksimumnya. Tujuannya adalah untuk melihat apakah ada data ekstrim per variabel data. Langkah berikutnya adalah melakukan preprocessing pada data. Tujuannya adalah untuk memfilter variabel apa saja yang akan dijadikan fitur berdasarkan korelasi heatmap, membuat label klasifikasi berdasarkan median MPG, menghapus data kosong maupun terduplikat, dan menyimpan semua hasil perubahan.

Langkah berikutnya adalah melakukan pemisahan data antara data latih dan data testing. Tujuannya adalah untuk melakukan pelatihan pada model ML menggunakan satu dataset yang sama. Langkah berikutnya adalah melatih Model dan melakukan evaluasi pada kinerja model. Fungsi nya adalah untuk melihat apakah performa dari model yang dilatih sudah akurat dan presisi atau belum. Langkah berikutnya adalah menguji model dengan data baru diluar dari dataset. Dan tahap terakhir adalah melakukan analisis pada hasil prediksi model.

3.2 PREPROCESSING DATA

Preprocessing Data merupakan teknik untuk membersihkan dan menyiapkan data mentah agar menjadi data yang siap untuk dipakai dalam melatih model ML. Langkah langkah yang dilakukan saat preprocessing data pada dataset yang peneliti gunakan adalah sebagai berikut :

```
import pandas as pd

import numpy as np

df_raw = pd.read_json(

    "X:\Code\Tubes_Phyton_Kel4\Dataset\Raw\cars2.json")

df = df_raw.copy()

baris_duplikat = df[df.duplicated()]

print("jumlah data duplikat adalah", baris_duplikat.shape)
```

```

print(df.isnull().sum())

df = df.rename(columns = {"Miles_per_Gallon":"MPG",
                          "Horsepower":"Tenaga Mesin",
                          "Weight_in_lbs":"Berat Dalam LBS", "Displacement":"Kapasitas
                          Mesin"})

df.head(5)

df = df[['MPG', 'Tenaga Mesin', 'Berat Dalam LBS', 'Kapasitas Mesin']]

df.dropna(inplace=True)

median_mpg = df['MPG'].median()

df['Irit atau Boros'] = np.where(df['MPG'] >= median_mpg, 1, 0)

df.to_csv("X:\Code\Tubes_Phyton_Kel4\Dataset\Processed\
        cars_processed.csv", index=False)

```

Penjelasan syntax yang digunakan pada kode ini adalah sebagai berikut :

- Syntax `df.duplicated()` berfungsi untuk mencari data yang terduplikat atau lebih dari satu pada satu dataset. Pada dataset yang digunakan, tidak terdapat data yang ganda.
- Syntax `df.isnull().sum()` berfungsi untuk mencari data yang kosong dan `.sum()` berfungsi untuk menjumlahkan semua data kosong yang sudah terdapat. Pada dataset yang digunakan, terdapat 14 data kosong yaitu 8 data MPG dan 6 data Horsepower.
- Syntax `df = df.rename(columns = {"Miles_per_Gallon":"MPG", "Horsepower":"Tenaga Mesin", "Weight_in_lbs":"Berat Dalam LBS", "Displacement":"Kapasitas Mesin"})` berfungsi untuk mengganti nama fitur menjadi nama yang lain.
- Syntax `df = df[['MPG', 'Tenaga Mesin', 'Berat Dalam LBS', 'Kapasitas Mesin']]` berfungsi untuk mengganti dataframe yang sebelumnya menggunakan keseluruhan data menjadi dataframe yang hanya berisi fitur yang digunakan yaitu MPG, Tenaga Mesin, Berat Dalam LBS, dan Kapasitas Mesin.

- Syntax `df.dropna(inplace=True)` berfungsi untuk menghapus data yang hilang secara keseluruhan termasuk seluruh baris dimana data kosong tersebut berada.
- Syntax `median_mpg = df['MPG'].median()` berfungsi untuk mencari nilai median dari data MPG.
- Syntax `df['Irit atau Boros'] = np.where(df['MPG'] >= median_mpg, 1, 0)` berfungsi untuk membuat label klasifikasi berdasarkan mediannya.
- Syntax `df.to_csv("X:\Code\Tubes_Phyton_Kel4\Dataset\Processed\cars_processed.csv", index=False)` berfungsi untuk merubah dataframe yang telah diolah ke bentuk csv.

3.3 PEMBAGIAN DATA

Pembagian data adalah teknik untuk melakukan pemecahan data menjadi data latihan dan data test untuk melatih model ML. Dari data tersebut, model akan melakukan perhitungan matematis dan mengetes dirinya sendiri dengan data yang telah dipecah sebelumnya. Proses melakukan pembagian data ini menggunakan library `sklearn train_test_split`. Random state merupakan pola pengambilan data secara random dimana dari seluruh populasi data akan diambil sampel sebagai data test yang akan digunakan untuk melatih model.

3.4 MODEL MACHINE LEARNING YANG DIGUNAKAN

Model yang digunakan pada penelitian ini adalah Logistic Regression dimana model ini adalah tipe yang paling cocok dengan analisis irit dan borosnya kendaraan karena logistic regression hanya melakukan klasifikasi biner (irit/boros) serta tidak memerlukan algoritma kompleks pada pengerjaannya.

Berikut ini adalah proses training pada model logistic regression :

```
import pandas as pd

import joblib

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score
```

```

df = pd.read_csv("X:\Code\Tubes_Phyton_Kel4\
                Dataset\Processed\cars_processed.csv")

X = df[['Tenaga Mesin', 'Berat Dalam LBS', 'Kapasitas Mesin']]

y = df['Irit atau Boros']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=1)

model = LogisticRegression(solver='liblinear')

model.fit(X_train, y_train)

accuracy = accuracy_score(y_test, model.predict(X_test))

print(f"Akurasi Model: {accuracy:.2f}")

joblib.dump(model, "model_logistic.joblib")

```

Penjelasan syntax dari yang digunakan pada kode ini adalah sebagai berikut :

- Syntax from sklearn.model_selection import train_test_split berfungsi untuk mengimport fitur train test split dari library sklearn.
- Syntax from sklearn.linear_model import LogisticRegression berfungsi untuk mengimport fitur algoritma logistic regression dari library sklearn.
- Syntax sklearn.metrics import accuracy_score berfungsi untuk mengimport fitur akurasi dari library sklearn.
- Syntax X = df[['Tenaga Mesin', 'Berat Dalam LBS', 'Kapasitas Mesin']] berfungsi untuk mendefinisikan Tenaga Mesin, Berat Dalam LBS, dan kapasitas Mesin sebagai variabel X.
- Syntax y = df['Irit atau Boros'] berfungsi untuk mendefinisikan Irit atau Boros sebagai variabel y
- Syntax X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1) berfungsi untuk membagi dataset menjadi data training dan data testing dengan pola pengambilan random 1.
- Syntax model = LogisticRegression(solver='liblinear') berfungsi untuk mendefinisikan logistic regression sebagai variabel model.
- Syntax model.fit(X_train, y_train) berfungsi untuk melakukan training pada model.

- Syntax `accuracy = accuracy_score(y_test, model.predict(X_test))` berfungsi untuk menilai akurasi berdasarkan variabel kunci `y_test` dengan prediksi model `X_test`.
- Syntax `joblib.dump(model, "model_logistic.joblib")` berfungsi untuk menyimpan hasil model yang telah ditraining kedalam library `joblib`.

BAB 4

HASIL DAN PEMBAHASAN

4.1 EVALUASI HASIL TRAINING MODEL

Setelah melakukan training, model akan digunakan untuk melakukan prediksi efisiensi bahan bakar menggunakan data test diluar dataset. Namun sebelum itu, diperlukan evaluasi kinerja model dalam memprediksi untuk memastikan tidak adanya bias pada data, overfitting, maupun underfitting. Confusion matrix merupakan salah satu cara untuk memeriksa perbandingan apakah model berhasil memprediksi benar dan dibandingkan dengan ground truth atau kunci jawabannya.

Berikut merupakan kode dari confusion matrikx :

```
from sklearn.metrics import confusion_matrix

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

df = pd.read_csv("X:\Code\Tubes_Phyton_Kel4\

    Dataset\Processed\cars_processed.csv")

X = df[['Tenaga Mesin', 'Berat Dalam LBS', 'Kapasitas Mesin']]

y = df['Irit atau Boros']

X_train, X_test, y_train, y_test = train_test_split(X, y,

    test_size=0.2, random_state=1)

model = LogisticRegression(solver='liblinear')

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

cm = confusion_matrix(y_test, y_pred)
```



```
plt.figure(figsize=(6, 5))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['BOROS', 'IRIT'], yticklabels=['BOROS', 'IRIT'])

plt.xlabel('Prediksi Model')

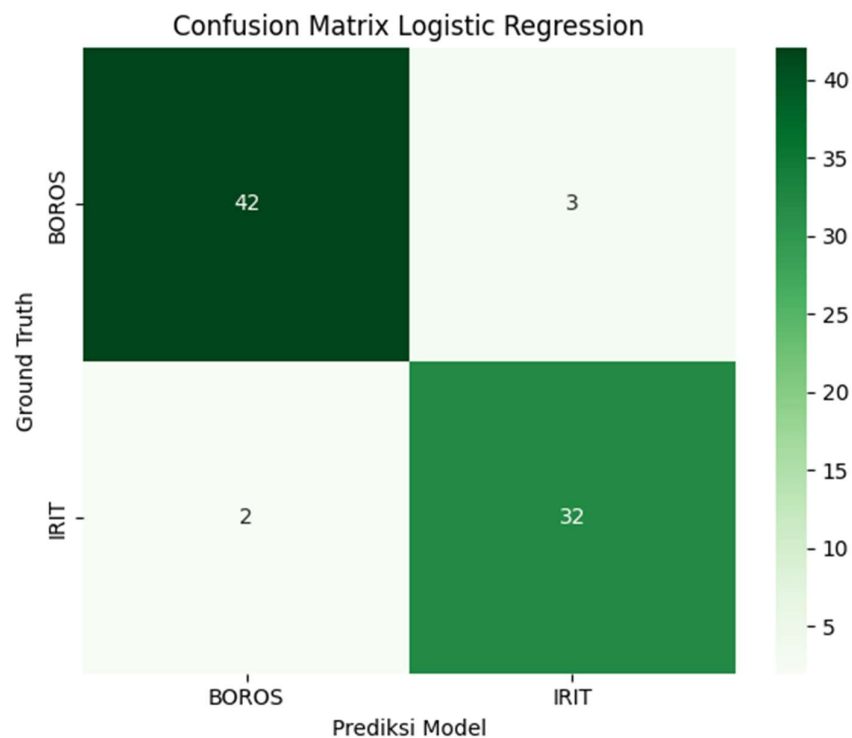
plt.ylabel('Ground Truth')

plt.title('Confusion Matrix Logistic Regression')

plt.tight_layout()

plt.show()
```

Output dari kode ini adalah sebagai berikut :



Gambar 4.1 Confusion Matrix

Berdasarkan visual dari gambar 4.1, terlihat bahwa model memiliki empat output perbandingan antara prediksi model dengan ground truth atau kunci jawabannya yang dapat disingkat menjadi True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Dari keempat output ini, dapat dicari evaluasi lain seperti akurasi, presisi, recall, F1 Score, dan bentuk evaluasi lainnya.

Maka, output dari confusion matrix bisa dituliskan sebagai berikut :

- Model memprediksi irit dan kunci jawabannya irit (TP) dengan score 32
- Model memprediksi boros dan kunci jawabannya boros (TN) dengan score 42
- Model memprediksi boros namun kunci jawabannya irit (FP) dengan score 2
- Model memprediksi irit namun kunci jawabannya boros (FN) dengan score 3

Rumus dari akurasi adalah sebagai berikut :

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{32 + 42}{32 + 42 + 2 + 3}$$

$$= \frac{74}{79}$$

$$Akurasi = 0,93$$

Berdasarkan perhitungan tersebut, model memiliki tingkat akurasi sebesar 0,93. Artinya prediksi yang dihasilkan oleh model 93% benar.

Rumus dari presisi model dalam menebak irit adalah sebagai berikut :

$$Presisi = \frac{TP}{TP + FP}$$

$$= \frac{32}{32 + 2}$$

$$Presisi = 0,94$$

Berdasarkan perhitungan tersebut, model memiliki tingkat presisi sebesar 0,94. Artinya setiap model memprediksi mobil irit, ketepatan menjawabnya 94% dan kemungkinan besar mobil tersebut memang irit. Dari kedua evaluasi tersebut, dapat disimpulkan bahwa model dapat memprediksi mobil boros atau irit secara akurat dan setiap tebakan irit memiliki presisi tinggi.

4.2 PREDIKSI DATA BARU

Setelah melakukan evaluasi model dan mendapat kesimpulan bahwa model tersebut akurat dan presisi, langkah berikutnya adalah memasukan data baru untuk memprediksi mobil yang ada. Pada penelitian ini, peneliti menggunakan 6 sampel mobil yang masing masing datanya sebagai berikut :

1. Toyota Fortuner 2.4 Diesel
 - Tenaga Mesin : 148 HP
 - Berat Kendaraan : 1.900 Kg
 - Kapasitas Mesin : 2.393 CC
2. Honda HR-V 1.5
 - Tenaga Mesin : 121 HP
 - Berat Kendaraan : 1.250 Kg
 - Kapasitas Mesin : 1.498 CC
3. Mitsubishi Pajero Sport 3.0
 - Tenaga Mesin : 220 HP
 - Berat Kendaraan : 2.100 Kg
 - Kapasitas Mesin : 3.000 CC
4. Toyota Alphard 3.5
 - Tenaga Mesin : 300 HP
 - Berat Kendaraan : 2.000 Kg
 - Kapasitas Mesin : 3.500 CC
5. Nissan Grand Livina
 - Tenaga Mesin : 107 HP
 - Berat Kendaraan : 1250 Kg
 - Kapasitas Mesin : 1.498 CC
6. Toyota Avanza 2015
 - Tenaga Mesin : 95 HP
 - Berat Kendaraan : 1.110 Kg
 - Kapasitas Mesin : 1.297 CC

Berikut merupakan kode untuk melakukan analisis :

```
import joblib

def pengubah_satuan(hp, berat_kg, cc):

    LBS = 2.20462

    CU_IN = 16.3871

    berat_lbs = berat_kg*LBS

    mesin_cu_in = cc/CU_IN

    return hp, berat_lbs, mesin_cu_in

def dataset():

    model = joblib.load("X:/Code/Tubes_Phyton_Kel4/Model ML/

        model_logistic.joblib")

    Nama = input("Masukkan Nama Kendaraan : ")

    hp = float(input("Masukkan horsepower kendaraan (hp): "))

    berat_kg = float(input("Masukkan berat kendaraan (kg): "))

    cc = float(input("Masukkan kapasitas mesin (cc): "))

    hp, berat_lbs, mesin_cu_in = pengubah_satuan(hp, berat_kg, cc)

    analisis = [[hp, berat_lbs, mesin_cu_in]]

    preds = model.predict(analisis)

    if preds[0] == 1:

        hasil = "IRIT"

    else:

        hasil = "BOROS"

    print("\n=== HASIL ANALISIS ===")

    print(f>Nama Mobil          : {Nama}")

    print(f>Horsepower          : {hp}")

    print(f>Berat Kendaraan      : {berat_lbs:.2f} lbs")
```

```

print(f"Kapasitas Mesin      : {mesin_cu_in:.2f} cu in")

print(f"Hasil Prediksi       : {hasil}")

return hasil

while True:

    print("\n=== MENU ===")

    print("1. Analisis Menggunakan Model Logistic Regression")

    print("0. Keluar")

    pilihan = input("Masukkan pilihan: ")

    if pilihan == "1":

        dataset()

    elif pilihan == "0":

        print("Program selesai.")

        break

    else:

        print("Pilihan tidak valid.")

```

Output dari kode ini adalah sebagai berikut :

```

=== HASIL ANALISIS ===
Nama Mobil      : Toyota Fortuner 2.4 Diesel
Horsepower      : 148.0
Berat Kendaraan : 4188.78 lbs
Kapasitas Mesin : 146.03 cu in
Hasil Prediksi  : IRIT

```

Gambar 4.2 Hasil Analisis Toyota Fortuner 2.4 Diesel

```

=== HASIL ANALISIS ===
Nama Mobil      : Honda HR-V 1.5
Horsepower      : 121.0
Berat Kendaraan : 2755.77 lbs
Kapasitas Mesin : 91.41 cu in
Hasil Prediksi  : IRIT

```

Gambar 4.3 Hasil Analisis Honda HR-V 1.5

```

=== HASIL ANALISIS ===
Nama Mobil      : Mitsubishi Pajero Sport 3.0
Horsepower      : 220.0
Berat Kendaraan : 4629.70 lbs
Kapasitas Mesin : 183.07 cu in
Hasil Prediksi  : BOROS

```

Gambar 4.4 Hasil Analisis Mitsubishi Pajero Sport 3.0

```

=== HASIL ANALISIS ===
Nama Mobil      : Toyota Alphard 3.5
Horsepower      : 300.0
Berat Kendaraan : 4409.24 lbs
Kapasitas Mesin : 213.58 cu in
Hasil Prediksi  : BOROS

```

Gambar 4.5 hasil Analisis Toyota Alphard 3.5

```

=== HASIL ANALISIS ===
Nama Mobil      : Nissan Grand Livina
Horsepower      : 107.0
Berat Kendaraan : 2755.77 lbs
Kapasitas Mesin : 91.41 cu in
Hasil Prediksi  : IRIT

```

Gambar 4.6 Hasil Analisis Nissan grand Livina

```
=== HASIL ANALISIS ===  
Nama Mobil      : Toyota Avanza 2015  
Horsepower      : 95.0  
Berat Kendaraan : 2447.13 lbs  
Kapasitas Mesin : 79.15 cu in  
Hasil Prediksi  : IRIT
```

Gambar 4.7 Hasil Analisis Toyota Avanza 2015

Dari hasil analisis keenam sampel mobil menggunakan model Regresi Logistik, dapat terlihat bahwa mobil dengan kapasitas mesin yang besar cenderung lebih boros dibandingkan dengan mobil berkapasitas mesin lebih kecil. Selain itu tenaga mesin dan berat juga memiliki andil dalam menentukan apakah mobil menjadi boros ataupun irit. Dapat disimpulkan bahwa rata rata mobil keluarga di indonesia tergolong irit dalam penggunaan bahan bakar.

BAB 5

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Berdasarkan hasil analisis dari enam sampel mobil keluarga di Indonesia menggunakan model Regresi Logistik, dapat disimpulkan bahwa pemilihan mobil dengan kapasitas mesin yang lebih tinggi akan menyebabkan penggunaan bahan bakar yang lebih banyak dibandingkan mobil dengan kapasitas mesin yang lebih rendah dan menyebabkan mobil tersebut menjadi boros. Pemilihan horsepower dan berat juga berpengaruh pada konsumsi bahan bakar, namun tidak sekuat kapasitas mesin.

Model Regresi Logistik terbukti efektif untuk melakukan klasifikasi irit dan borosnya mobil. Meskipun terdapat beberapa data yang ekstrim dan menjauh dari rata-rata ketika dilihat berdasarkan standar deviasi, model tetap berhasil menangkap gambaran penting pada pola penggunaan bahan bakar di satu dataset tersebut dengan akurasi sebesar 0,93 dan presisi menebak irit sebesar 0,94.

5.2 SARAN

Berikut ini adalah saran yang bisa peneliti berikan :

1. Untuk masyarakat yang peduli lingkungan dan sedang mencari mobil dengan konsumsi bahan bakar sedikit, disarankan untuk memilih mobil dengan kapasitas mesin yang rendah seperti Nissan Grand Livina, dan Toyota Avanza.
2. Untuk produsen mobil yang ada di Indonesia, disarankan untuk lebih banyak membuat mobil yang ramah lingkungan dan mulailah beralih ke kendaraan listrik demi lingkungan yang lebih sehat.
3. Untuk peneliti yang akan melanjutkan penelitian ini, masih banyak variabel dan fitur yang dapat ditambahkan untuk memperkuat model. Variabel kecil seperti konsumsi bahan bakar di jalan tol dan di jalan biasa, kemacetan pada jalan raya, dan karakteristik pengguna kendaraan dapat mempengaruhi perilaku model dalam menangkap pola di sebuah dataset saat pelatihan.

DAFTAR PUSTAKA

- Sklearn
- Matplotlib
- Pandas
- Numpy
- Joblib