

**TUGAS PENGANTI UTS**  
**DATA SAINS DAN ANALISIS**  
**ANALISIS TERHADAP VEHICLE CO2 EMISSIONS DATASET**



**Disusun Oleh:**

Muhammad Naufal Afif (1103210089)

Fachrurozi (1103210194)

**PROGRAM STUDI S1 TEKNIK KOMPUTER**  
**FAKULTAS TEKNIK ELEKTRO**  
**UNIVERSITAS TELKOM**  
**BANDUNG**  
**2024**

## A. Dataset

- Nama Dataset: Vehicle CO2 Emissions Dataset
- Sumber dataset: <https://www.kaggle.com/datasets/brsahan/vehicle-co2-emissions-dataset/data>

## B. Deskripsi Dataset

Dataset ini disusun untuk mendukung pendekatan Regresi Linier Sederhana (SLR) dan Regresi Linier Berganda (MLR) untuk proyek pembelajaran mesin. Dataset ini berisi informasi tentang spesifikasi kendaraan, konsumsi bahan bakar, dan emisi CO2, yang dikumpulkan untuk menganalisis dampak kendaraan terhadap lingkungan dan memperkirakan emisi CO2 menggunakan model regresi.

## C. Penjelasan fitur

- Brand: Merek atau manufaktur kendaraan (misalnya, Toyota, Ford, BMW).
- Vehicle Type: Klasifikasi kendaraan berdasarkan ukuran dan penggunaannya (misalnya, SUV, Sedan).
- Engine Size (L): Volume kapasitas mesin dalam liter.
- Cylinders: Jumlah silinder yang ada pada mesin kendaraan.
- Transmission: Jenis transmisi yang digunakan (misalnya, Otomatis, Manual).
- Fuel Type: Jenis bahan bakar yang digunakan oleh kendaraan  
X : Bensin biasa  
Z : Bensin Premium  
D : Diesel  
E : Ethanol  
N : Gas
- Fuel Consumption (City, Hwy, and Combined): Efisiensi bahan bakar yang diukur dalam liter per 100 kilometer (L/100 km).
- CO2 Emissions (g/km): Emisi karbon dioksida per kilometer (variabel target untuk prediksi).

## D. Library

Library yang digunakan adalah :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

## E. Pengolahan Data

### 1) Load Dataset

Source Code:

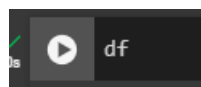
```
df= pd.read_csv('/content/drive/MyDrive/co2.csv')
```

Penjelasan :

Source code ini menginput file 'co2.csv' dari dalam google drive ke dalam google colab lalu disimpan di variabel df.

### 2) Preview Data

Source Code:



Output:

	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel type	Fuel Consumption City (L/100 km)	Fuel Consumption Hwy (L/100 km)	Fuel Consumption Comb (L/100 km)
0	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	6.7	8.5
1	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7.7	9.6
2	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	5.8	5.9
3	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	9.1	11.1
4	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8.7	10.6
...	...	...	...	...	...	...	...	...	...	...
7380	VOLVO	XC40 T5 AWD	SUV - SMALL	2.0	4	AS8	Z	10.7	7.7	9.4
7381	VOLVO	XC60 T5 AWD	SUV - SMALL	2.0	4	AS8	Z	11.2	8.3	9.9
7382	VOLVO	XC60 T6 AWD	SUV - SMALL	2.0	4	AS8	Z	11.7	8.6	10.3
7383	VOLVO	XC90 T5 AWD	SUV - STANDARD	2.0	4	AS8	Z	11.2	8.3	9.9
7384	VOLVO	XC90 T6 AWD	SUV - STANDARD	2.0	4	AS8	Z	12.2	8.7	10.7

7385 rows x 12 columns

Penjelasan:

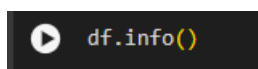
Source code ini menampilkan isi dari 'co2.csv' yang sudah dimasukkan kedalam variabel df.

## F. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) merupakan tahap awal dalam menganalisis dataset yang bertujuan untuk memperoleh pemahaman mendalam mengenai karakteristik data, baik secara statistik deskriptif maupun melalui visualisasi. Tujuan utama EDA adalah untuk mengidentifikasi pola, hubungan antar variabel, serta mendeteksi adanya anomali yang dapat mempengaruhi hasil analisis, sekaligus memeriksa kualitas data. EDA sangat penting karena membantu dalam memahami struktur data, mengidentifikasi kebutuhan preprocessing, serta merumuskan hipotesis untuk analisis lanjutan.

### 1) Informasi Data

Source Code:



Output:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7385 entries, 0 to 7384
Data columns (total 12 columns):
 #   Column                                          Non-Null Count  Dtype
---  -
 0   Make                                           7385 non-null   object
 1   Model                                          7385 non-null   object
 2   Vehicle Class                                 7385 non-null   object
 3   Engine Size(L)                               7385 non-null   float64
 4   Cylinders                                     7385 non-null   int64
 5   Transmission                                 7385 non-null   object
 6   Fuel Type                                     7385 non-null   object
 7   Fuel Consumption City (L/100 km)             7385 non-null   float64
 8   Fuel Consumption Hwy (L/100 km)              7385 non-null   float64
 9   Fuel Consumption Comb (L/100 km)             7385 non-null   float64
10   Fuel Consumption Comb (mpg)                  7385 non-null   int64
11   CO2 Emissions(g/km)                          7385 non-null   int64
dtypes: float64(4), int64(3), object(5)
memory usage: 692.5+ KB

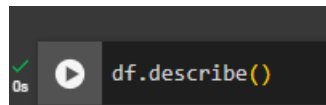
```

### Penjelasan:

Code ini digunakan untuk memberikan informasi ringkas tentang struktur DataFrame, seperti nama dan tipe data kolom, jumlah nilai yang tersedia di setiap kolom, serta penggunaan memori.

## 2) Statistik Deskriptif Data Numerik

### Source Code:



```
df.describe()
```

### Output:

	Engine Size(L)	Cylinders	Fuel Consumption City (L/100 km)	Fuel Consumption Hwy (L/100 km)	Fuel Consumption Comb (L/100 km)	Fuel Consumption Comb (mpg)	CO2 Emissions(g/km)
count	7385.000000	7385.000000	7385.000000	7385.000000	7385.000000	7385.000000	7385.000000
mean	3.160068	5.615030	12.556534	9.041706	10.975071	27.481652	250.584699
std	1.354170	1.828307	3.500274	2.224456	2.892506	7.231879	58.512679
min	0.900000	3.000000	4.200000	4.000000	4.100000	11.000000	96.000000
25%	2.000000	4.000000	10.100000	7.500000	8.900000	22.000000	208.000000
50%	3.000000	6.000000	12.100000	8.700000	10.600000	27.000000	246.000000
75%	3.700000	6.000000	14.600000	10.200000	12.600000	32.000000	288.000000
max	8.400000	16.000000	30.600000	20.600000	26.100000	69.000000	522.000000

### Penjelasan:

df.describe() digunakan untuk menghasilkan statistik deskriptif dari kolom numerik dalam DataFrame, seperti jumlah data, rata-rata, deviasi standar, nilai minimum, kuartil, dan nilai maksimum. Fungsi ini membantu memahami distribusi dan variasi data.

## 3) Analisis Korelasi Data Numerik

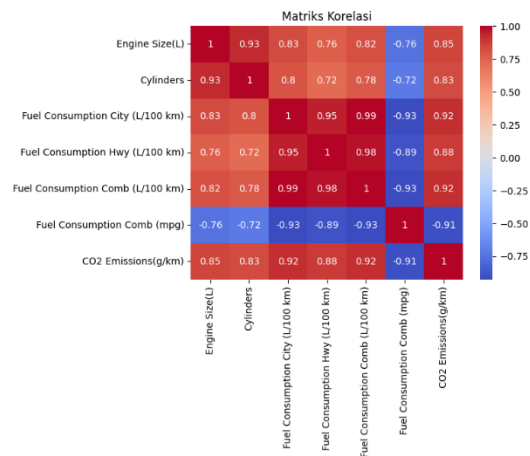
### Source Code:

```

correlation_matrix = df.select_dtypes(include=['int', 'float']).corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Matriks Korelasi')
plt.show()

```

### Output:



### Penjelasan:

Kode ini digunakan untuk menampilkan heatmap yang menunjukkan korelasi antar kolom numerik dalam Data Frame.

## G. Visualisasi Data

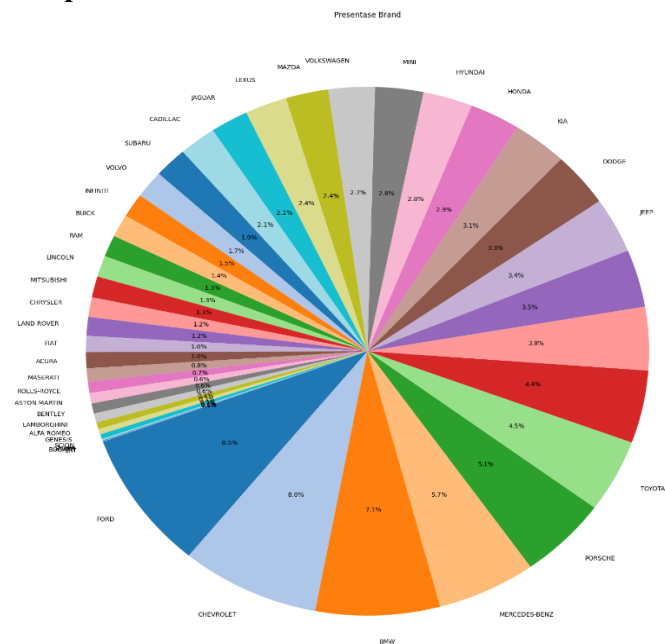
### 1) Pie Chart

#### Source Code:

```
# Menghitung jumlah masing-masing brand
brand_counts = df['Brand'].value_counts()

# Membuat pie chart
plt.figure(figsize=(40, 20))
brand_counts.plot.pie(autopct='%1.1f%%', startangle=200, colors=plt.cm.tab20.colors)
plt.title('Presentase Brand')
plt.ylabel('') # Menghilangkan label sumbu Y
plt.show()
```

#### Output:



### Penjelasan:

Kode ini digunakan untuk menampilkan presentase Brand(manufaktur) menggunakan pie chart.

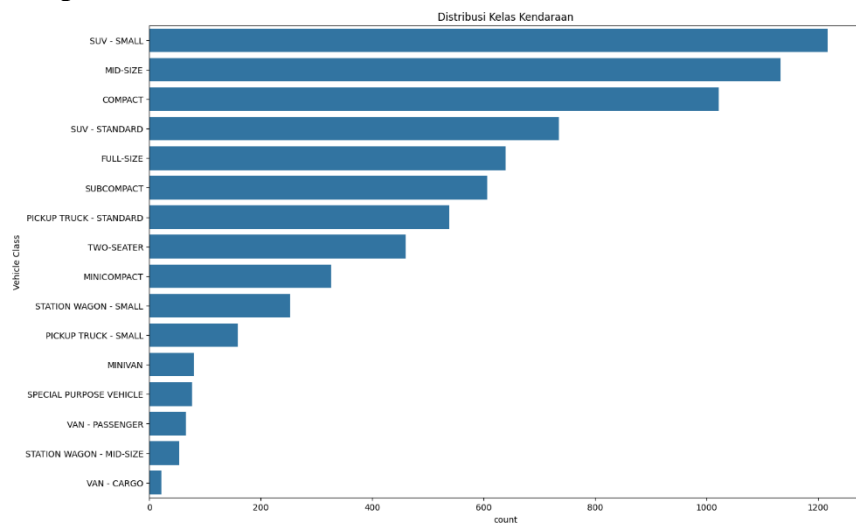
## 2) Bar Chart

- **Distribusi Kelas Kendaraan**

Source Code:

```
plt.figure(figsize=(15, 10))
sns.countplot(y='Vehicle Class', data=df, order=df['Vehicle Class'].value_counts().index)
plt.title('Distribusi Kelas Kendaraan')
plt.show()
```

Output:

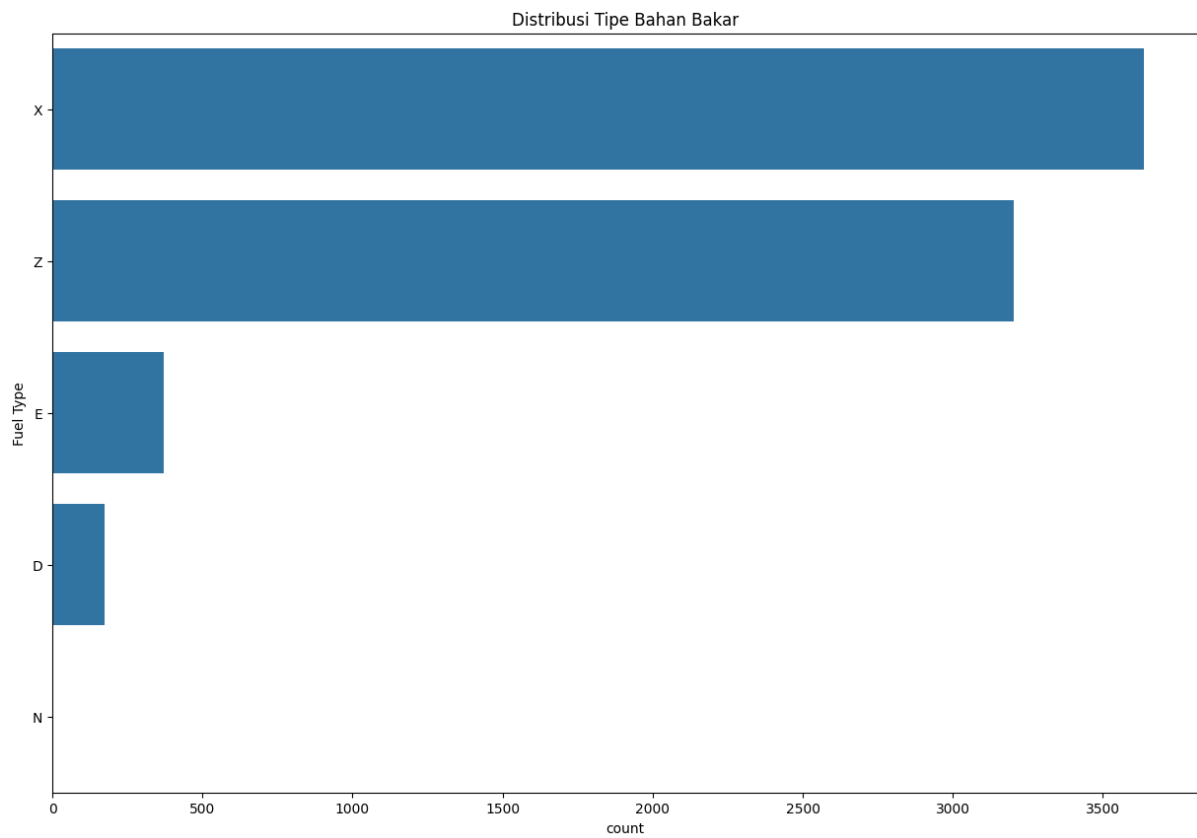


- **Distribusi tipe bahan bakar**

Source Code:

```
plt.figure(figsize=(15, 10))
sns.countplot(y='Fuel Type', data=df, order=df['Fuel Type'].value_counts().index)
plt.title('Distribusi Tipe Bahan Bakar')
plt.show()
```

Output:

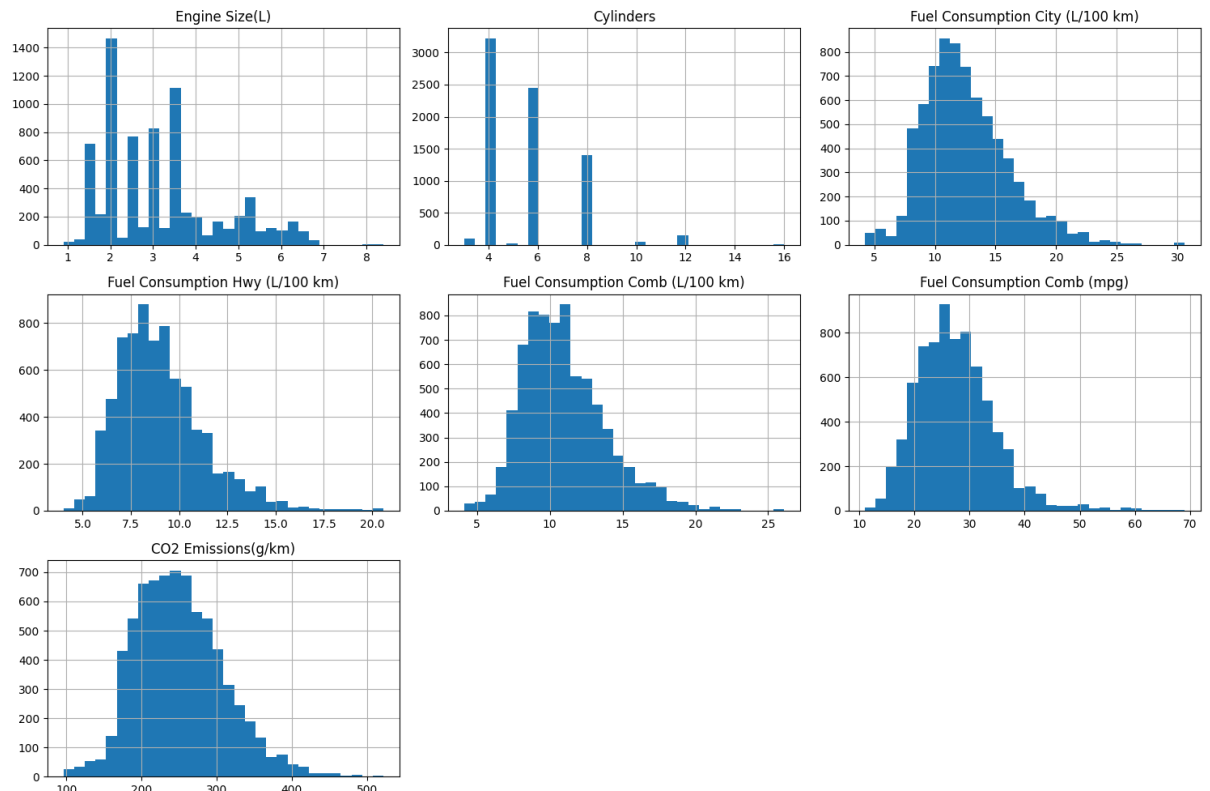


### 3) Histogram

Source Code:

```
[98] df.hist(bins=30, figsize=(15, 10))  
plt.tight_layout()  
plt.show()
```

Output:



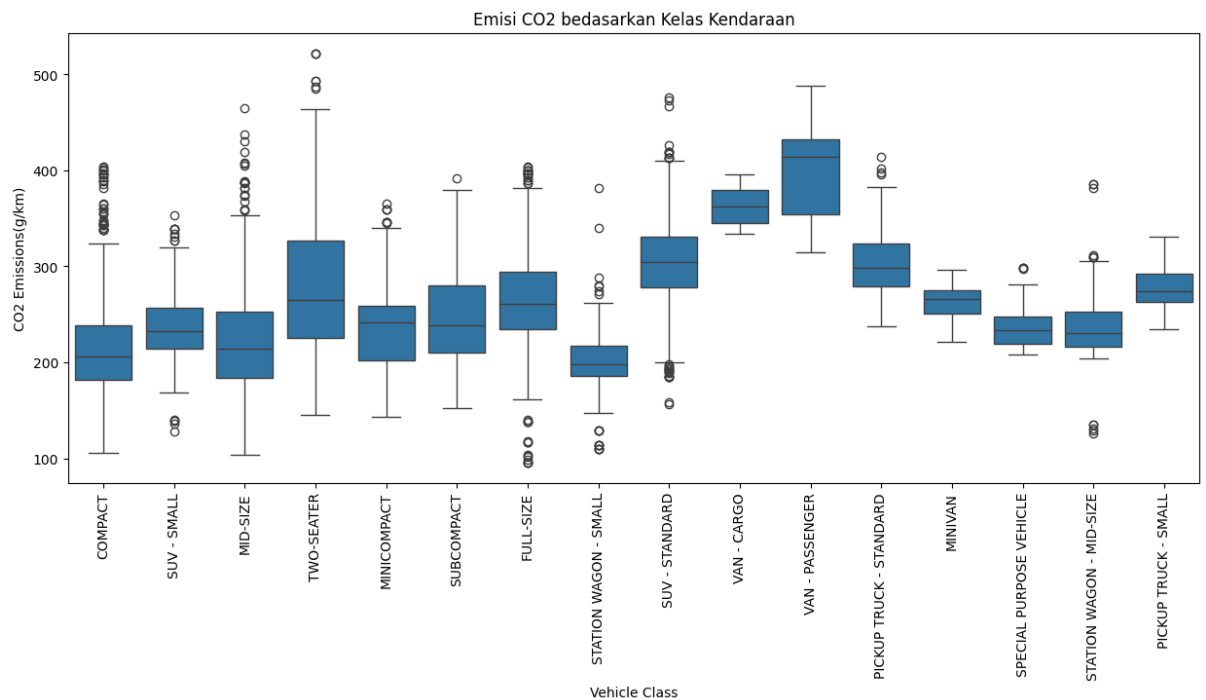
#### 4) Boxplot

Source Code:

```
plt.figure(figsize=(15, 6))
sns.boxplot(x='Vehicle Class', y='CO2 Emissions(g/km)', data=df)
plt.xticks(rotation=90)
plt.title('Emisi CO2 berdasarkan Kelas Kendaraan')
plt.show()
```

Output:





### Penjelasan:

Ini digunakan untuk menampilkan Emisi CO2 yang di keluarkan berdasarkan Kelas kendaraan dalam bentuk boxplot.

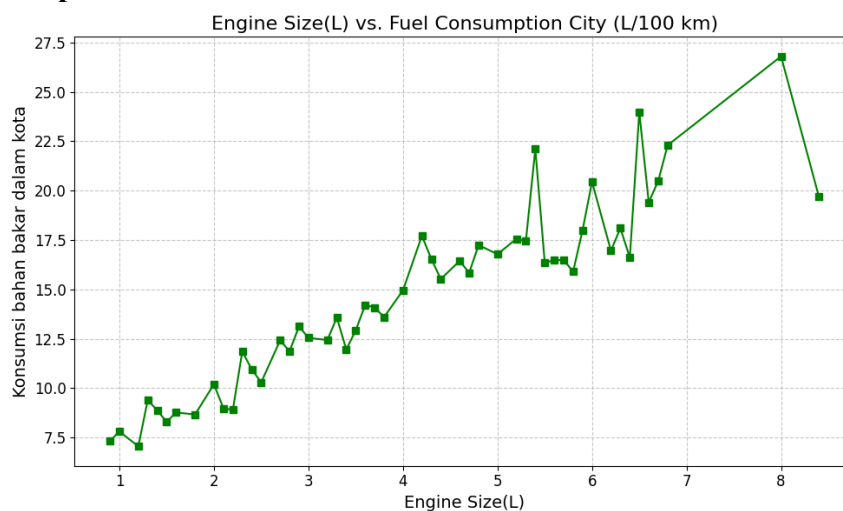
## 5) Line Graph

### Source Code:

```
# Menghitung rata-rata Sleep Duration berdasarkan Physical Activity Level
avg_Fuel_Consumption_City= df.groupby('Engine Size(L)')['Fuel Consumption City (L/100 km)'].mean()

# Membuat Line Graph
plt.figure(figsize=(10, 6))
plt.plot(avg_Fuel_Consumption_City.index, avg_Fuel_Consumption_City.values, marker='s', linestyle='--', color='green')
plt.title('Engine Size(L) vs. Fuel Consumption City (L/100 km)', fontsize=16)
plt.xlabel('Engine Size(L)', fontsize=14)
plt.ylabel('Konsumsi bahan bakar dalam kota', fontsize=14)
plt.grid(linestyle='--', alpha=0.7)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.tight_layout()
plt.show()
```

### Output:



### Penjelasan :

Berikut adalah Konsumsi bahan bakar dalam kota berdasarkan ukuran Mesin menggunakan Line Chart

## H. Data Preperation

### 1) Menampilkan Duplikat

#### Source Code:

```
df[df.duplicated(keep=False)].style.set_properties(**{'background-color':'#2a9d8f','color':'white','border':'2.5px, solid black'})
```

#### Output:

531	INFINITI	Q60 CONVERTIBLE	SUBCOMPACT	3.700000	6	AS7	Z	13.600000	9.300000	11.7000
532	INFINITI	Q60 CONVERTIBLE	SUBCOMPACT	3.700000	6	M6	Z	14.300000	9.800000	12.3000
535	INFINITI	Q70	MID-SIZE	3.700000	6	AS7	Z	12.800000	8.900000	11.0000
537	INFINITI	Q70 AWD	MID-SIZE	3.700000	6	AS7	Z	13.200000	9.600000	11.6000
547	JAGUAR	F-TYPE CONVERTIBLE	TWO-SEATER	3.000000	6	AS8	Z	11.800000	8.400000	10.3000
548	JAGUAR	F-TYPE S CONVERTIBLE	TWO-SEATER	3.000000	6	AS8	Z	12.200000	8.700000	10.6000
549	JAGUAR	F-TYPE V8 S CONVERTIBLE	TWO-SEATER	5.000000	8	AS8	Z	15.000000	10.200000	12.8000

### 2) Menampilkan Jumlah data duplikat

#### Source Code:

```
df.duplicated().sum()
```

#### Output:

1103

### 3) Menghapus data duplikat

#### Source Code:

```
df = df.drop_duplicates()
```

#### Hasil :

```
df.duplicated().sum()
```

0

### 4) Menampilkan nilai yang hilang

#### Source Code:

```
df.isna().sum()
```

#### Output:

	0
Brand	0
Model	0
Vehicle Class	0
Engine Size(L)	0
Cylinders	0
Transmission	0
Fuel Type	0
Fuel Consumption City (L/100 km)	0
Fuel Consumption Hwy (L/100 km)	0
Fuel Consumption Comb (L/100 km)	0
Fuel Consumption Comb (mpg)	0
CO2 Emissions(g/km)	0
dtype: int64	

### Penjelasan:

Berikut adalah untuk menampilkan nilai null atau nilai yang hilang pada data set. Pada dataset ini tidak ada nilai yang hilang

## I. Hasil Akhir

df.head(15)											
	Brand	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption City (L/100 km)	Fuel Consumption Hwy (L/100 km)	Fuel Consumption Comb (L/100 km)	
0	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	6.7	8.5	
1	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7.7	9.6	
2	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	5.8	5.9	
3	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	9.1	11.1	
4	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8.7	10.6	
5	ACURA	RLX	MID-SIZE	3.5	6	AS6	Z	11.9	7.7	10.0	
6	ACURA	TL	MID-SIZE	3.5	6	AS6	Z	11.8	8.1	10.1	
7	ACURA	TL AWD	MID-SIZE	3.7	6	AS6	Z	12.8	9.0	11.1	
8	ACURA	TL AWD	MID-SIZE	3.7	6	M6	Z	13.4	9.5	11.6	
9	ACURA	TSX	COMPACT	2.4	4	AS5	Z	10.6	7.5	9.2	
10	ACURA	TSX	COMPACT	2.4	4	M6	Z	11.2	8.1	9.8	
11	ACURA	TSX	COMPACT	3.5	6	AS5	Z	12.1	8.3	10.4	
12	ALFA ROMEO	4C	TWO-SEATER	1.8	4	AM6	Z	9.7	6.9	8.4	
13	ASTON MARTIN	DB9	MINICOMPACT	5.9	12	A6	Z	18.0	12.6	15.6	
14	ASTON MARTIN	RAPIDE	SUBCOMPACT	5.9	12	A6	Z	18.0	12.6	15.6	

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)