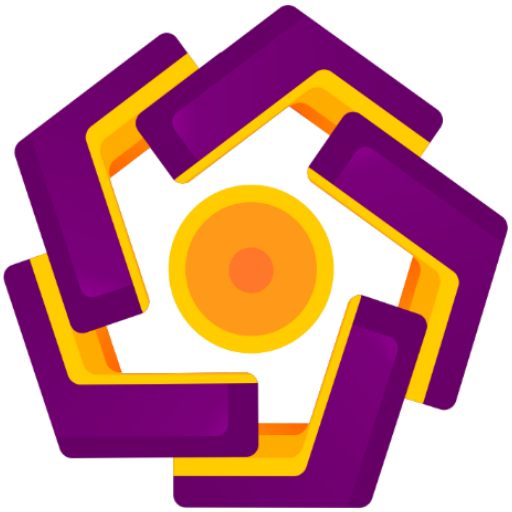
BIG DATA & PREDICTIVE ANALYTICS
FINAL PROJECT: EMISI KARBON DIOKSIDA YANG DIHASILKAN OLEH MOBIL PADA NEGARA CANADA

Dosen Pengampu : Arif Dwi Laksito, M.Kom



Nama Kelompok:

1. Haikal Raditya Fadhilah (21.11.3910)
2. Wulan Kristiyanti (21.11.3924)
3. Widdia Glory Anggrenny (21.11.3936)
4. Gilang Ramadhani (21.11.3946)

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA

2023

Daftar Isi

1 Latar Belakang

Kanada adalah salah satu negara maju dengan ekonomi yang berkembang pesat dan memiliki populasi yang besar. Seperti banyak negara lainnya, Kanada juga menghadapi tantangan dalam mengelola emisi karbon dioksida dan dampak perubahan iklim. Emisi karbon dioksida (CO2) adalah salah satu gas rumah kaca utama yang bertanggung jawab atas pemanasan global dan perubahan iklim yang sedang terjadi. Pentingnya prediksi pada Objek Emisi Karbon Dioksida di Kanada:

a. Pengambilan Kebijakan yang Efektif: Prediksi emisi karbon dioksida di Kanada menjadi sangat penting untuk menginformasikan pembuatan kebijakan lingkungan yang efektif. Dengan memiliki pemahaman yang baik tentang tren emisi di masa depan, pemerintah dan pemangku kepentingan dapat merancang langkah-langkah yang tepat untuk mengurangi emisi CO2 dan mencapai target iklim yang telah ditetapkan.

b. Perencanaan Infrastruktur dan Investasi: Prediksi emisi membantu dalam perencanaan infrastruktur dan investasi jangka panjang. Misalnya, informasi tentang pertumbuhan emisi di sektor transportasi akan membantu mengarahkan investasi pada transportasi berkelanjutan, seperti pengembangan kendaraan listrik atau sistem transportasi publik yang lebih efisien.

c. Identifikasi Tantangan dan Peluang: Prediksi emisi karbon dioksida juga membantu mengidentifikasi tantangan dan peluang yang terkait dengan penerapan kebijakan perubahan iklim. Dengan memahami sektor dan wilayah yang menyumbang pada emisi tinggi, langkah-langkah khusus dapat diambil untuk mengatasi masalah tersebut dan memanfaatkan peluang yang ada.

Manfaat dari Prediksi Emisi Karbon Dioksida:

a.Responsibilitas dan Akuntabilitas: Prediksi emisi membantu meningkatkan responsibilitas dan akuntabilitas pemerintah dan industri terkait dalam mencapai target pengurangan emisi.

b. Evaluasi Kebijakan dan Program: Prediksi emisi memungkinkan evaluasi keberhasilan kebijakan dan program perubahan iklim yang ada. Jika prediksi menunjukkan bahwa target tidak akan tercapai, maka langkah-langkah perbaikan dapat diambil untuk memastikan keberhasilan di masa mendatang.

c. Kerjasama Internasional: Dalam kerangka perubahan iklim global, prediksi emisi memainkan peran penting dalam memfasilitasi kerjasama internasional. Negara-negara lain dapat menggunakan data prediksi ini untuk memahami kontribusi Kanada dalam mengatasi perubahan iklim secara global dan merumuskan kerjasama yang lebih efektif.

Secara keseluruhan, prediksi emisi karbon dioksida di Kanada sangat penting untuk membantu mengatasi perubahan iklim dan mencapai target pengurangan emisi. Eksperimen yang dilakukan dalam konteks ini bertujuan untuk meningkatkan keakuratan dan efektivitas prediksi serta memberikan panduan bagi pengambilan kebijakan yang lebih baik dan berkelanjutan.

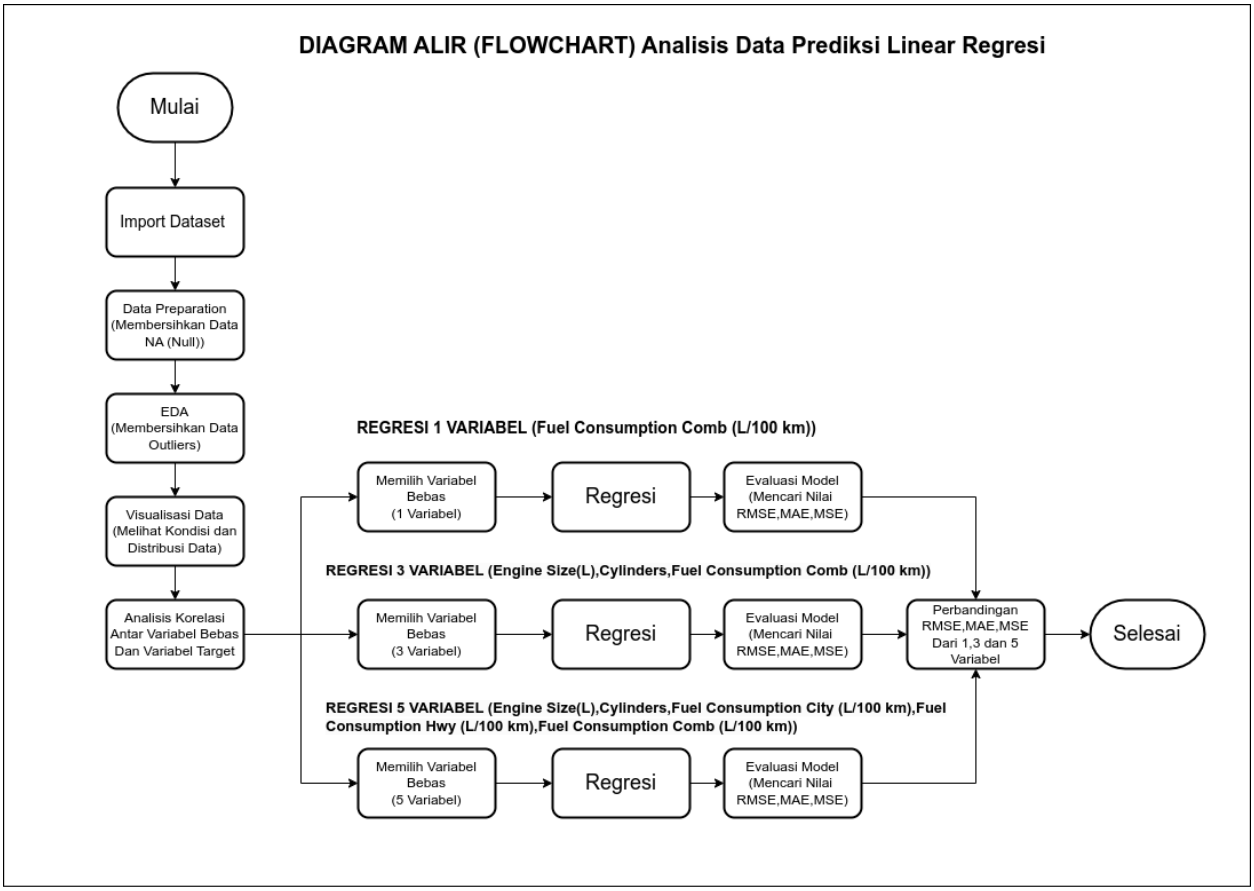
2 Metode

2.1 Dataset

Dataset yang kelompok kami gunakan berasal dari github, link URL : https://github.com/HaikalRFadhilahh/fp-BigDataPA/blob/master/DATASET/CO2%20Emissions_Canada.csv, dengan jumlah baris 7386 dan jumlah kolom 12, dan jenis kolomnya meliputi :

Make : Merk mobil yang beroperasi di negara Canada
Model : Tipe kendaraan
Vahicle Class : Kelas kendaraan
Engine Size : Ukuran mesin
Cylinders: Jumlah silinder mesin pada kendaraan
Transmitions : Jenis transmisi
Fuel Type : Jenis bahan bakar
Fuel Consumtion City (L/100km) : Jumlah pemakaian bahan bakar di kota
Fuel Consumption Comb (L/100 km) : Jumlah pemakaian bahan bakar di jalan raya (L/100km)
Fuel Consumption Comb (mpg) : Jumlah pemakaian bahan bakar di jalan raya (mpg)
CO2 Emissions(g/km) : Jumlah Karbon Dioksida yang dihasilkan oleh kendaraan

2.2 Alur penelitian



Gambar diagram alir proses prediksi regresi jumlah emisi karbondioksida

Keterangan : Menurut diagram alir diatas sebelum melakukan regresi kami melakukan pembersihan data terlebih dahulu yaitu pembersihan data null maupun data ouliers di variabel independent yang akan menjadi variabel bebas untuk regresi agar hasil regresi Root Mean Square Error,Mean Absolute Error dan Mean Square Error dapat lebih akurat.

3 Eksperimen

LIBRARY

Agar memudahkan dalam melakukan analisis data serta mencari linear regresi,kami menggunakan beberapa library yang telah di buat oleh orang lain.
Berikut library yang di gunakan beserta fungsinya :

- Pandas berfungsi untuk membaca dan mengolah dataset
- Matplotlib berfungsi untuk membuat visualisasi data
- Seaborn berfungsi untuk membuat visualisasi Heatmap korelasi
- Spicy berfungsi untuk menghitung nilai korelasi pearson
- Numpy berfungsi untuk menghitung RMSE (Root Mean Square Error)
- Sklearn linear_model berfungsi untuk membuat Linear Regresi
- Sklearn metrics berfungsi untuk menghitung MAE (Mean Absolute Error) Dan MSE (Mean Square Error)
- Sklearn model_selection berfungsi untuk membagi dataset menjadi data train dan data test

IMPORT LIBRARY

```
[ ]
1 import pandas as pd
2 from matplotlib import pyplot as plt
3 import seaborn as sns
4 from scipy import stats
5 import numpy as np
6 from sklearn.linear_model import LinearRegression
7 from sklearn.metrics import mean_absolute_error, mean_squared_error,mean_squared_error
8 from sklearn.model_selection import train_test_split
```

Gambar code import library python

VARIABLE

1. Variable Dependent : CO2 Emissions(g/km). Ini menjadi variabel yang ingin kita prediksi atau pahami faktor-faktornya.

2. Variable Independent : Vahicle Class, Engine Size, Cylinders, Transmitions, Fuel Type, Fuel Consumtion City (L/100km), Fuel Consumption Comb (L/100 km), Fuel Consumption Comb (mpg). Ini menjadi variabel yang diasumsikan memiliki pengaruh terhadap variabel dependen .

EDA (Exploratory Data Analysis)

a. Menampilkan 10 baris teratas dari datasets
df.head(10)

Out[599…

	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)	Fuel Consumption Comb (mpg)	Emis
0	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	6.7	8.5	33	
1	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7.7	9.6	29	
2	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	5.8	5.9	48	
3	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	9.1	11.1	25	
4	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8.7	10.6	27	
5	ACURA	RLX	MID-SIZE	3.5	6	AS6	Z	11.9	7.7	10.0	28	
6	ACURA	TL	MID-SIZE	3.5	6	AS6	Z	11.8	8.1	10.1	28	
7	ACURA	TL AWD	MID-SIZE	3.7	6	AS6	Z	12.8	9.0	11.1	25	
8	ACURA	TL AWD	MID-SIZE	3.7	6	M6	Z	13.4	9.5	11.6	24	
9	ACURA	TSX	COMPACT	2.4	4	AS5	Z	10.6	7.5	9.2	31	

b. Perintah df.info() akan menampilkan beberapa informasi tentang DataFrame, antara lain: Jumlah baris dan kolom dalam DataFrame,Tipe data dari setiap kolom,Jumlah nilai non-null (non-kosong) dalam setiap kolom,Penggunaan memori (memori yang dialokasikan untuk DataFrame).

In [602…

```
df.info()
```

```
<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
```

c. Melihat deskripsi dari dataset
Bertujuan untuk memahami karakteristik dan struktur dari dataset yang akan digunakan dalam analisis atau pemodelan. Deskripsi dataset memberikan gambaran tentang

bagaimana data terorganisir, informasi tentang variabel, dan statistik dasar yang terkait dengan masing-masing variabel.

In [603…
```
df.describe()
```

Out[603…

	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

d. Membersihkan Data Outliers

Membersihkan data outliers adalah langkah dalam pra-pemrosesan data yang bertujuan untuk mengidentifikasi dan mengatasi nilai-nilai ekstrem atau outlier dalam dataset. Outliers adalah nilai-nilai yang jauh berbeda dari sebagian besar data dan dapat mempengaruhi hasil analisis dan pemodelan jika tidak ditangani dengan benar.

In [604…
```
#Membuat Fungsi Untuk Membersihkan Outliers
def clear_outliers (dataset,kolom):
    Q1 = dataset[kolom].quantile(0.25)
    Q3 = dataset[kolom].quantile(0.75)
    IQR = Q3 - Q1
    min_iqr = Q1 - 1.5 * IQR;
    max_iqr = Q3 + 1.5 * IQR;
    return dataset.loc[(dataset[kolom] >= min_iqr) & (dataset[kolom] <= max_iqr)].reset_index(drop=True)
```

In [605…
```
# Membersikan Outliers Kolom Fuel Consumption City (L/100 km)
df = clear_outliers(df,'Fuel Consumption City (L/100 km)')
```

In [606…
```
# Membersikan Outliers Kolom Fuel Consumption Hwy (L/100 km)
df = clear_outliers(df,'Fuel Consumption Hwy (L/100 km)')
```

In [607…
```
# Membersikan Outliers Kolom Fuel Consumption Comb (L/100 km)
df = clear_outliers(df,'Fuel Consumption Comb (L/100 km)')
```

In [608…
```
# Membersikan Outliers Kolom Fuel Consumption Comb (mpg)
df = clear_outliers(df,'Fuel Consumption Comb (mpg)')
```

In [609…
```
# Mengurutkan Data Berdasarkan Kolom C02 Emissions Secara Ascending Serta Mereset Index
df.sort_values('CO2 Emissions(g/km)',ascending=True,inplace=True)
df.reset_index(inplace=True,drop=True)
df
```

e. Melihat Korelasi Setiap Variabel

Memahami hubungan atau asosiasi antara variabel-variabel dalam dataset. Korelasi mengukur derajat hubungan linier antara dua variabel numerik. Korelasi dapat membantu mengidentifikasi pola dan wawasan penting dalam data serta memberikan gambaran tentang seberapa kuat dan arah hubungan antara variabel-variabel tersebut.

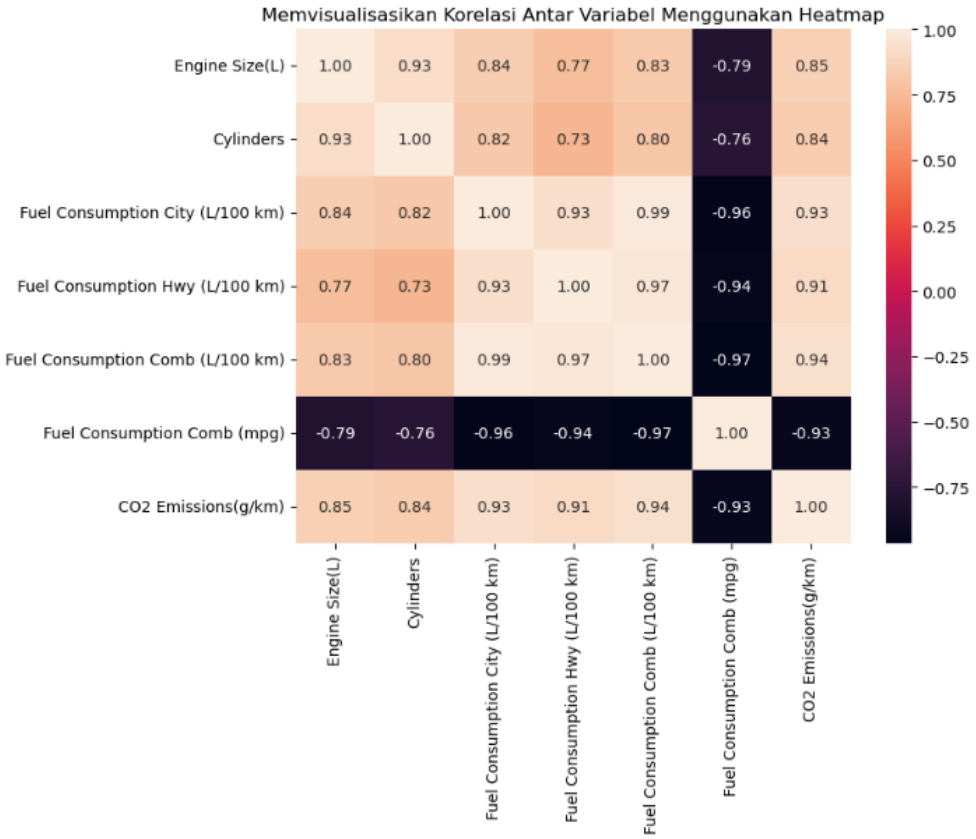
Out[611…

	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)
Engine Size(L)	1.000000	0.926442	0.844804	0.766804	0.830888	-0.792541	0.854056
Cylinders	0.926442	1.000000	0.822533	0.729232	0.802957	-0.756637	0.835555
Fuel Consumption City (L/100 km)	0.844804	0.822533	1.000000	0.933860	0.992294	-0.958631	0.933247
Fuel Consumption Hwy (L/100 km)	0.766804	0.729232	0.933860	1.000000	0.970506	-0.940306	0.909434
Fuel Consumption Comb (L/100 km)	0.830888	0.802957	0.992294	0.970506	1.000000	-0.967054	0.939334
Fuel Consumption Comb (mpg)	-0.792541	-0.756637	-0.958631	-0.940306	-0.967054	1.000000	-0.930364
CO2 Emissions(g/km)	0.854056	0.835555	0.933247	0.909434	0.939334	-0.930364	1.000000

ANALISIS KORELASI

a. Memvisualisasikan Semua korelasi menggunakan Heatmap dengan library Seaborn dan Matplotlib

Dengan memilih variabel yang cocok untuk regresi di perlukan analisis korelasi yang baik,Visualisasi sangat di perlukan untuk melihat seberapa baik korelasi antar 2 variabel numeric dengan menggunakan warna kita bisa lebih teliti dalam mencari sebuah korelasi



Gambar Heatmap korelasi antar variable numeric

Menurut Heatmap diatas dapat menjadi sebuah acuan untuk mencari korelasi antar variabel yang baik,korelasi yang baik merupakan korelasi yang memiliki nilai mendekati 1 ataupun -1.Visualisasi Heat menunjukkan bawah jika warna semakin gelap atau semakin putih maka korelasi mendekati sempurna.

b. Membuat Fungsi Pengecekan Korelasi Serta Nilai Korelasi Pearson

Untuk mendapatkan variabel yang saling berkorelasi kuat denga akurat kita perlu mengecek kembali dengan menggunakan fungsi pearsonr dari library scipy,pada fungsi yang kami buat kami pegelompokan seperti berikut :

- Jika nilai korelasi pearson lebih besar dari 0.7 atau lebih kecil dari -0.7 maka korelasi tersebut merupakan korelasi kuat.

- Jika korelasi pearson lebih besar dari 0.5 atau lebih kecil dari -0.5 maka korelasi tersebut merupakan korelasi sedang.
- Jika korelasi tersebut lebih kecil dari 0.5 dan lebih besar dari dari -0.5 maka korelasi tersebut merupakan korelasi lemah.

In [84]:
```
def cek_korelasi(dataset=df,kolomDependen='CO2 Emissions(g/km)',arrKolomIndependent=['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)',]):
    for data in arrKolomIndependent:
        pearson_corr,p_value = stats.pearsonr(dataset[kolomDependen],dataset[data])

        if data == kolomDependen:
            print(f"Dataset Kolom '{kolomDependen}' Dan '{data}' Memiliki Korelasi Pearson Sempurna Di Karenakan Kedua Kolom
            print("")
        elif pearson_corr > 0.7:
            print(f"Dataset Kolom '{kolomDependen}' Dan '{data}' Berkorelasi Positif Kuat Dengan Nilai Korelasi Pearson : {p
            print("")
        elif pearson_corr > 0.5:
            print(f"Dataset Kolom '{kolomDependen}' Dan '{data}' Berkorelasi Positif Sedang Dengan Nilai Korelasi Pearson :
            print("")
        elif pearson_corr > 0:
            print(f"Dataset Kolom '{kolomDependen}' Dan '{data}' Berkorelasi Positif Lemah Dengan Nilai Korelasi Pearson : {
            print("")
        elif pearson_corr == 0:
            print(f"Dataset Kolom '{kolomDependen}' Dan '{data}' Tidak Memilik Korelasi Dengan Nilai Korelasi Pearson : {pea
            print("")
        elif pearson_corr < -0.7:
            print(f"Dataset Kolom '{kolomDependen}' Dan '{data}' Berkorelasi Negatif Kuat Dengan Nilai Korelasi Pearson : {p
            print("")
        elif pearson_corr < -0.5:
            print(f"Dataset Kolom '{kolomDependen}' Dan '{data}' Berkorelasi Negatif Sedang Dengan Nilai Korelasi Pearson :
            print("")
        elif pearson_corr < 0:
            print(f"Dataset Kolom '{kolomDependen}' Dan '{data}' Berkorelasi Negatif Lemah Dengan Nilai Korelasi Pearson : {
            print("")
```

Gambar deklarasi fungsi untuk menghitung nilai korelasi pearson

Dengan membuat fungsi seperti diatasi kita dapat mencari nilai pearson dan jenis korelasinya tanpa harus membuat code berulang-ulang.

4.Mengecek Korelasi Kolom 'C02 Emissions' dengan Semua Kolom Numeric

In [85]:
```
cek_korelasi(kolomDependen='CO2 Emissions(g/km)')
```

```
Dataset Kolom 'CO2 Emissions(g/km)' Dan 'Engine Size(L)' Berkorelasi Positif Kuat Dengan Nilai Korelasi Pearson : 0.85405649840
03576

Dataset Kolom 'CO2 Emissions(g/km)' Dan 'Cylinders' Berkorelasi Positif Kuat Dengan Nilai Korelasi Pearson : 0.8355545081017088

Dataset Kolom 'CO2 Emissions(g/km)' Dan 'Fuel Consumption City (L/100 km)' Berkorelasi Positif Kuat Dengan Nilai Korelasi Pears
on : 0.9332467579756798

Dataset Kolom 'CO2 Emissions(g/km)' Dan 'Fuel Consumption Hwy (L/100 km)' Berkorelasi Positif Kuat Dengan Nilai Korelasi Pearso
n : 0.909433640624198

Dataset Kolom 'CO2 Emissions(g/km)' Dan 'Fuel Consumption Comb (L/100 km)' Berkorelasi Positif Kuat Dengan Nilai Korelasi Pears
on : 0.9393339672156101

Dataset Kolom 'CO2 Emissions(g/km)' Dan 'Fuel Consumption Comb (mpg)' Berkorelasi Negatif Kuat Dengan Nilai Korelasi Pearson :
-0.9303644465314539
```

5.Mengecek Korelasi Kolom 'Fuel Consumption Comb (L/100 km)' dengan Kolom Engine Size Dan Cylinders

In [86]:
```
cek_korelasi(kolomDependen='Fuel Consumption Comb (L/100 km)',arrKolomIndependent=['Engine Size(L)','Cylinders'])
```

```
Dataset Kolom 'Fuel Consumption Comb (L/100 km)' Dan 'Engine Size(L)' Berkorelasi Positif Kuat Dengan Nilai Korelasi Pearson :
0.8308883872989032

Dataset Kolom 'Fuel Consumption Comb (L/100 km)' Dan 'Cylinders' Berkorelasi Positif Kuat Dengan Nilai Korelasi Pearson : 0.802
9570802135054
```

Gambar cara penggunaan fungsi serta hasil output yang di hasilkan

REGRESI

Untuk melakukan regresi ada beberapa tahap yang harus di lakukan untuk mendapatkan hasil yang akurat.

Berikut langkah-langkah dalam Regresi :

a. Menentukan variabel bebas serta target atau variabel dependen yang akan di cari :

i. Menggunakan 1 variabel bebas (Independent) :

```
X = df[['Fuel Consumption Comb (L/100 km)']] #Mendeklarasikan Variabel X sebagai variabel bebas (Konsumsi B
y = df['CO2 Emissions(g/km)'] #Mendeklarasikan Variabel y sebagai variabel terikat (Emisi Gas Karbon Dioksi
```

Gambar deklarasi 1 variabel bebas dan 1 variabel target (Dependent)

ii. Menggunakan 3 variabel bebas (Independent) :

```
variabelBebas = ['Engine Size(L)','Cylinders','Fuel Consumption Comb (L/100 km)'] # Menggunakan Variabel Uk
X = df[variabelBebas]
y = df['CO2 Emissions(g/km)']
```

Gambar deklarasi 3 variable bebas dan 1 variabel target

iii. Menggunakan 5 variabel bebas (Independent) :

```
variabelBebas = ['Engine Size(L)','Cylinders','Fuel Consumption City (L/100 km)','Fuel Consumption Hwy (L/1
X = df[variabelBebas]
y = df['CO2 Emissions(g/km)']
```

Gambar deklarasi 5 variabel bebas dan 1 variabel target

b. Membagi dataset menjadi 2 bagian yaitu data train dan data test :

```
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.30, random_state=100);
```

Gambar pembagian data train dan data test

Agar dapat mendapatkan regresi yang baik kami melakukan pembagian data train sebanyak 70% dan data test sebanyak 30% dengan random state 100.dengan data testing sebanyak 30% sudah cukup untuk menguji model yang telah di buat.

c. Membuat model regresi dari library Sklearn :

```
linreg_model = LinearRegression()
linreg_model.fit(X_train,y_train)
```

Gambar mendeklarasikan serta membuat model prediksi dari data train

Dalam membuat mode prediksi Linear Regression kita menggunakan fungsi yang telah di ada dari library Sklearn.Dan untuk membuat model prediksi kita harus memasukkan data train yang telah kita bagi agar model dapat menghitung prediksi secara akurat.

d. Menghitung nilai koefisien dan intercept dari model :

i. Koefisien dan Intercept Value 1 variabel :

```
In [627…
print("Nilai Koefisien : ",linreg_model.coef_)
print("Nilai Intercept : ",linreg_model.intercept_)
```

```
Nilai Koefisien :  [20.62718779]
Nilai Intercept :  26.60625137646707
```

Gambar Nilai Koefisien dan Intercept 1 Variabel Independent

ii. Koefisien dan Intercept Value 3 variabel :

```
In [635…
print("Nilai Koefisien : ",linreg_model.coef_)
print("Nilai Intercept : ",linreg_model.intercept_)
```

```
Nilai Koefisien :  [ 4.65041075  4.01193518 16.11978529]
Nilai Intercept :  38.414137548029316
```

Gambar Nilai Koefisien dan Intercept 3 Variabel Independent

iii. Koefisien dan Intercept Value 5 variabel :

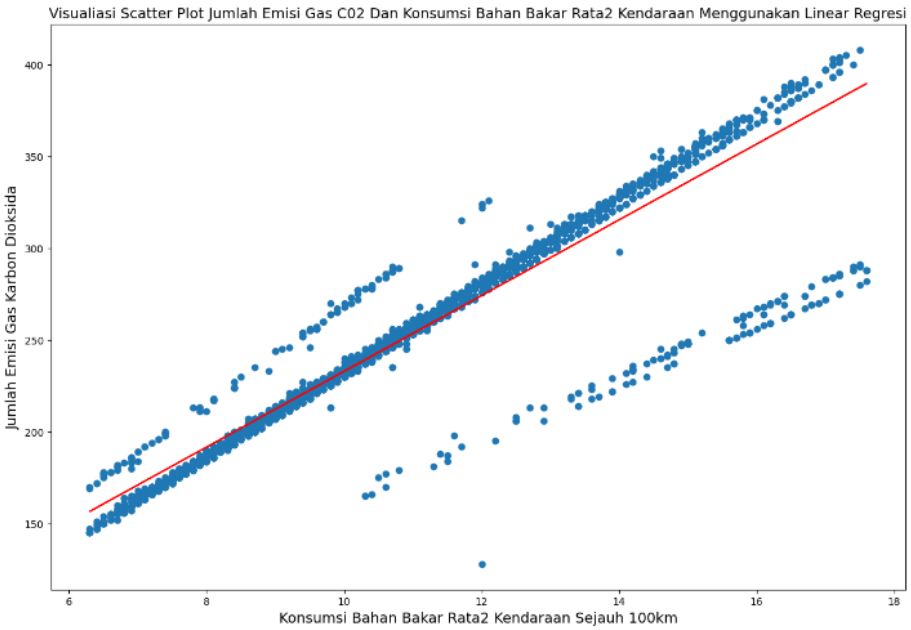
```
In [643…
print("Nilai Koefisien : ",linreg_model.coef_)
print("Nilai Intercept : ",linreg_model.intercept_)
```

```
Nilai Koefisien :  [ 4.95083014  4.74196803  0.07750876  6.05575548 10.98229415]
Nilai Intercept :  34.18025263070868
```

Gambar Nilai Koefisien dan Intercept 5 Variabel Independent

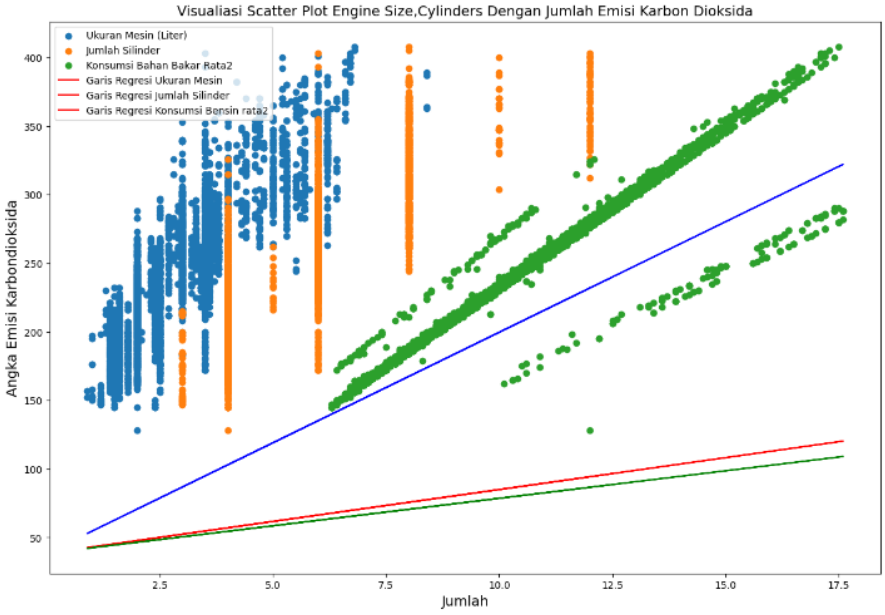
e. Visualisasi regresi dengan scatter plot :

i. Visualisasi Regresi 1 variabel :



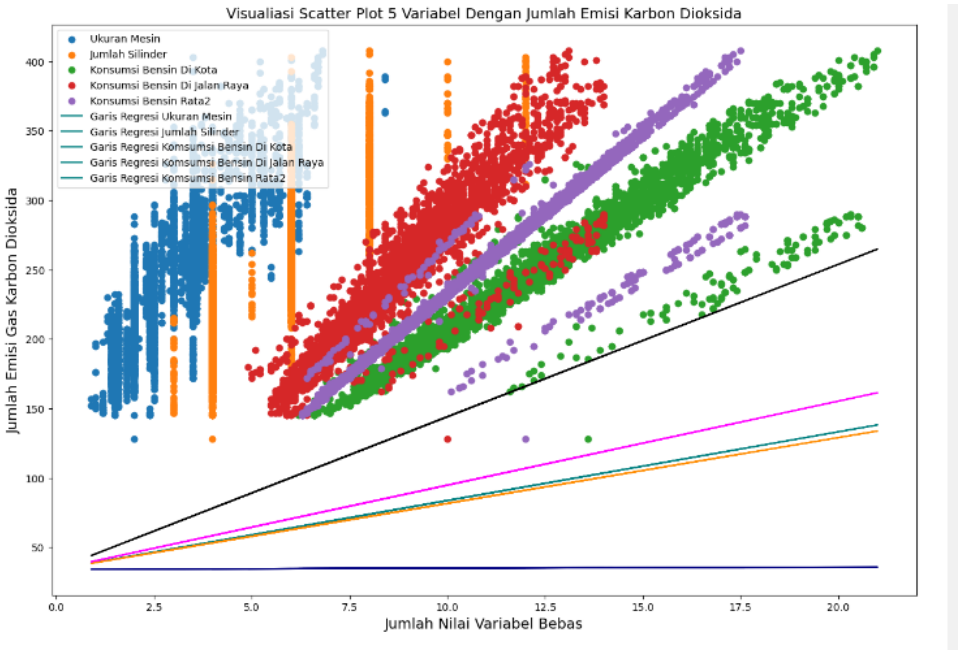
Gambar Visualisasi linear regresi 1 variabel

ii. Visualisasi Regresi 3 variabel :



Gambar Visualisasi linear regresi 3 variabel

iii. Visualisasi Regresi 5 variabel :



Gambar Visualisasi linear regresi 5 variabel

f. Melakukan prediksi data target dari data test :

```
y_pred = linreg_model.predict(X_test)
```

Gambar deklarasi variable prediksi dari variabel split testing

g. Melakukan evaluasi model RMSE,MAE dan MSE :

i. Evaluasi Model 1 Variabel :

```
In [630…
print("Root Mean Square Error (RMSE) : ",np.sqrt(mean_squared_error(np.array(y_test),y_pred)))
print("Mean Absolute Error (MAE): " ,mean_absolute_error(y_test,y_pred))
print("Mean Squared Error (MSE): " ,mean_squared_error(y_test,y_pred))

Root Mean Square Error (RMSE) :  18.81158402185771
Mean Absolute Error (MAE):  9.114851353897137
Mean Squared Error (MSE):  353.87569341141227
```

Gambar evaluasi model 1 variabel independent

ii. Evaluasi Model 3 Variabel :

```
In [638…
print("Root Mean Square Error (RMSE) : ",np.sqrt(mean_squared_error(np.array(y_test),y_pred)))
print("Mean Absolute Error (MAE): " ,mean_absolute_error(y_test,y_pred))
print("Mean Squared Error (MSE): " ,mean_squared_error(y_test,y_pred))

Root Mean Square Error (RMSE) :  16.429440573024458
Mean Absolute Error (MAE):  9.421002009136643
Mean Squared Error (MSE):  269.92651754254223
```

Gambar evaluasi model 3 variabel independent

iii. Evaluasi Model 5 Variabel :

```
In [646…
print("Root Mean Square Error (RMSE) : ",np.sqrt(mean_squared_error(np.array(y_test),y_pred)))
print("Mean Absolute Error (MAE): " ,mean_absolute_error(y_test,y_pred))
print("Mean Squared Error (MSE): " ,mean_squared_error(y_test,y_pred))

Root Mean Square Error (RMSE) :  16.345370226102396
Mean Absolute Error (MAE):  9.402089786842936
Mean Squared Error (MSE):  267.1711278283547
```

Gambar evaluasi model 5 variabel independent

Dengan menggunakan metode RMSE,MAE dan MSE kita dapat mengevaluasi model yang kita buat untuk memprediksi data target.dari metode evaluasi tersebut kita dapat mengetahui jumlah dan total error yang di hasilkan dari model dan dari hasil itulah kita dapat melihat keakuratan dari model prediksi yang kita buat

4 Hasil dan Evaluasi

a. Jika kita menggunakan linear regresi dengan 1 variabel bebas maka kita akan mendapatkan hasil 18.81158402185771.
b. Jika kita menggunakan linear regresi dengan 3 variabel bebas maka kita akan mendapatkan hasil 16.429440573024458
c. Jika kita menggunakan linear regresi dengan 5 variabel bebas maka kita akan mendapatkan hasil 16.345370226102396

5 Kesimpulan

Dapat diambil kesimpulan bahwa untuk dua perbandingan antara 1, 3, dan 5 variabel RMSE yang terbaik adalah dari 5 perbandingan yaitu 16.345370226102396 angka ini lebih kecil artinya model lebih baik dalam mendekati nilai sebenarnya dan memiliki kinerja prediksi yang lebih baik

6 Kontribusi anggota

Jelaskan kontribusi dari masing-masing anggota secara detail.

Nama Anggota Kelompok	Kontribusi Anggota Kelompok
1. Haikal Raditya Fadhilah (21.11.3910)	-Mencari datasets -Membuat EDA

	-Menyusun laporan -Menyusun PPT
2. Gilang Ramadhani (21.11.3946)	-Menyusun laporan -Menyusun PPT -Membuat Regresi Linear
3. Wulan Kristiyanti (21.11.3924)	-Menyusun laporan -Menyusun PPT - Membuat Korelasi
4. Widdia Glory Anggrenny (21.11.3936)	-Menyusun laporan -Menyusun PPT -Memasukkan Library

7 Lampiran

1. Link dataset

https://github.com/HaikalRFadhilahh/fp-BigDataPA/blob/master/DATASET/CO2%20Emissions_Canada.csv

2. Code python

https://github.com/HaikalRFadhilahh/fp-BigDataPA.git

3. Vidio presentasi

https://drive.google.com/file/d/1zPeJgWUMlM6B_bX8GT_l9qPxLlqaKF4c/view