Tim GCD

<table>
<tr><td>Judul</td><td colspan="2">Prediksi Harga Penjualan Mobil dengan Dataset "Automobile Dataset" Menggunakan Metode Regresi</td></tr>
<tr><td>Tujuan</td><td colspan="2">Membantu perusahaan otomotif (mobil) dalam menentukan harga jual yang paling cocok untuk sebuah mobil berdasarkan fitur-fitur tertentu yang ditawarkan</td></tr>
<tr><td>Sumber Data</td><td colspan="2">https://www.kaggle.com/datasets/toramky/automobile-dataset</td></tr>
<tr><td>Fitur Independen</td><td colspan="2">make, fuel-type, aspiration, num-of-doors, body-style, drive-wheels, engine-location, wheel-base, length, width, height, curb-weight, engine-type, num-of-cylinders, engine-size, fuel-system, bore, stroke, compression-ratio, horse-power, peak-rpm, city-mpg, highway-mpg</td></tr>
<tr><td>Fitur Dependen</td><td colspan="2">price/ harga</td></tr>
<tr><td>Ukuran Dataset</td><td colspan="2">206 baris data, 26 kolom fitur</td></tr>
<tr><td rowspan="2">Preprocessing</td><td>Membuang baris data yang hanya berisi “?”</td><td>

```
# delete row that contain any "?"

for i in raw_data.columns:
  raw_data = raw_data[raw_data[i] != '?']

print(raw_data.shape)
```

```
(159, 26)
```

</td></tr>
<tr><td>Memperbaiki data bertipe string yang seharusnya bernilai numerik (int/ float)</td><td>

```
# transform string numeric data to integer

for i in raw_data.columns:
  if raw_data[i].dtype == 'object':
    try:
      if "." in str(raw_data[i]):
        raw_data[i] = raw_data[i].astype('float64')
      else:
        raw_data[i] = raw_data[i].astype('int64')
    except:
      continue

raw_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 159 entries, 3 to 204
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symboling          159 non-null    int64
 1   normalized-losses  159 non-null    float64
 2   make               159 non-null    object
 3   fuel-type          159 non-null    object
 4   aspiration         159 non-null    object
 5   num-of-doors       159 non-null    object
 6   body-style         159 non-null    object
 7   drive-wheels       159 non-null    object
 8   engine-location    159 non-null    object
 9   wheel-base         159 non-null    float64
 10  length             159 non-null    float64
 11  width              159 non-null    float64
 12  height             159 non-null    float64
 13  curb-weight        159 non-null    int64
 14  engine-type        159 non-null    object
 15  num-of-cylinders   159 non-null    object
 16  engine-size        159 non-null    int64
 17  fuel-system        159 non-null    object
 18  bore               159 non-null    float64
 19  stroke             159 non-null    float64
 20  compression-ratio  159 non-null    float64
 21  horsepower         159 non-null    float64
 22  peak-rpm           159 non-null    float64
 23  city-mpg           159 non-null    int64
 24  highway-mpg        159 non-null    int64
 25  price              159 non-null    float64
dtypes: float64(11), int64(5), object(10)
```

</td></tr>
</table>

<table>
<tr>
<td></td>
<td>Memilih dan mempertahankan fitur-fitur terbaik untuk di-training</td>
<td>

```
new_data = pd.DataFrame({
    "merek": raw_data["make"],
    # "bahan-bakar": raw_data["fuel-type"],
    # "udara": raw_data["aspiration"],
    "jlh-pintu": raw_data["num-of-doors"],
    # "body": raw_data["body-style"],
    # "jenis-ban": raw_data["drive-wheels"],
    "roda": raw_data["wheel-base"],
    # "panjang": raw_data["length"],
    "lebar": raw_data["width"],
    "berat": raw_data["curb-weight"],
    # "jlh-silinder": raw_data["num-of-cylinders"],
    # "tipe-mesin": raw_data["engine-type"],
    "uk-mesin": raw_data["engine-size"],
    "tenaga": raw_data["horsepower"],
    # "rasio-comp": raw_data["compression-ratio"],
    "price": raw_data["price"],
})
new_data.head()
```

	merek	jlh-pintu	roda	lebar	berat	uk-mesin	tenaga	price
3	audi	four	99.8	66.2	2337	109	102.0	13950.0
4	audi	four	99.4	66.4	2824	136	115.0	17450.0
6	audi	four	105.8	71.4	2844	136	110.0	17710.0
8	audi	four	105.8	71.4	3086	131	140.0	23875.0
10	bmw	two	101.2	64.8	2395	108	101.0	16430.0

</td>
</tr>
<tr>
<td>Pembagian Train-Test</td>
<td colspan="2">

```
# memisahkan variabel x dan y dari tabel
X_data = new_data.drop('price', axis = 1)
y_data = new_data['price']
```

```
X_data = pd.get_dummies(X_data, drop_first = True, dtype = 'int8')
```

```
# bagi data untuk training n testing dgn rasio 8:2

X_train, X_test, y_train, y_test = train_test_split(
    X_data,
    y_data,
    test_size = 0.20,
    random_state = 50
)
```

</td>
</tr>
<tr>
<td>Pemilihan Model</td>
<td colspan="2">Regression Tree

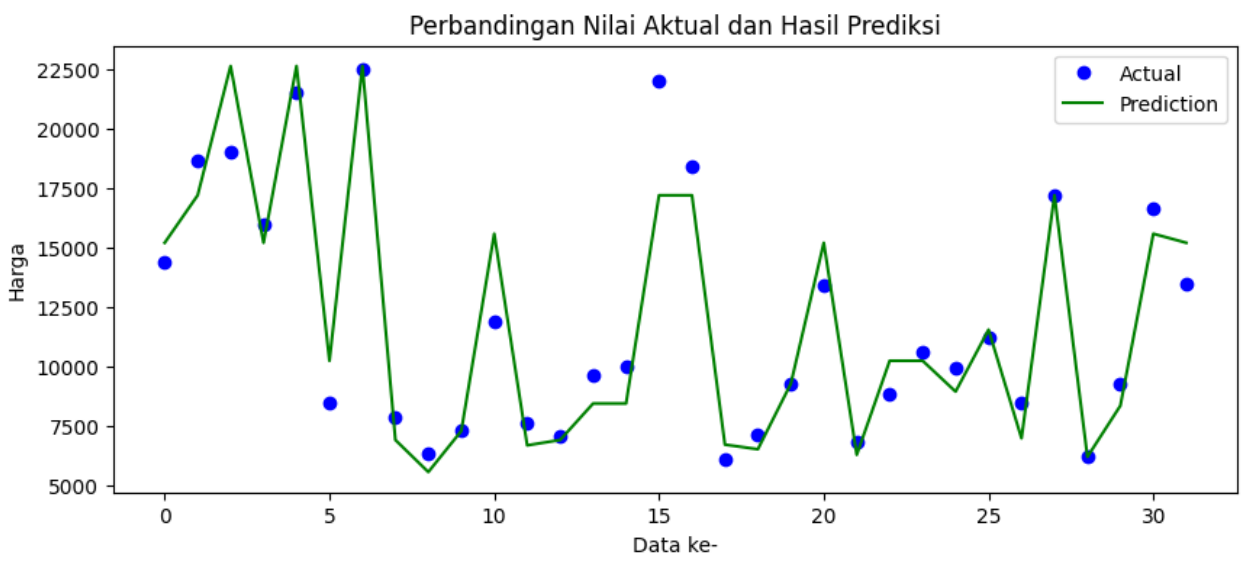
```
regtree_model = DecisionTreeRegressor(max_depth=10)
regtree_model.fit(X_train, y_train)

# membuat prediksi model
y_pred_regtree = regtree_model.predict(X_test)
```

</td>
</tr>
<tr>
<td>Evaluasi</td>
<td colspan="2">

```
# Evaluasi model regression tree - R^2, MSE, RMSE, MAPE
print("R^2  =", regtree_model.score(X_test, y_test))
print("MSE  =", metrics.mean_squared_error(y_test, y_pred_regtree))
print("RMSE =", np.sqrt(metrics.mean_squared_error(y_test, y_pred_regtree)))
print("MAPE =", metrics.mean_absolute_percentage_error(y_test, y_pred_regtree) * 100, "%")
```

```
R^2  = 0.901427271806723
MSE  = 2486555.6328125
RMSE = 1576.8816166131496
MAPE = 9.656007165029706 %
```

</td>
</tr>
</table>

Perbandingan Nilai Aktual dan Prediksi	## Visualisasi Nilai Aktual dengan Nilai Hasil Prediksi `[ ]` ``` plt.figure(figsize=(10, 4)) plt.plot(np.array(y_test), 'bo', label='Actual') plt.plot(y_pred_regtree, color="green", label='Prediction') plt.title('Perbandingan Nilai Aktual dan Hasil Prediksi') plt.xlabel('Data ke-') plt.ylabel('Harga') plt.legend(loc='upper right') plt.show() ``` 
Link PPT	https://www.canva.com/design/DAGSg8kgNKI/7Z-quaZPQTRCY3cFJFaZDA/edit?utm_content=DAGSg8kgNKI&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton
Link Notebook	https://colab.research.google.com/drive/160VmtZVjVjsP9zZbO7XJ9_tSV3QHM-Ci?usp=sharing