

Peramalan Jumlah Kendaraan Listrik di *State* Washington Menggunakan Analisis Runtun Waktu ARIMA

Team Statset

Diki Wahyudi | Jefta Adriel Heryadi | Kamal Muftie Yafi

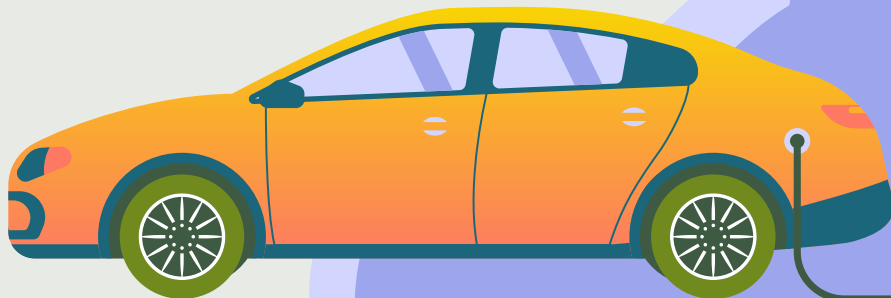


TABLE OF CONTENTS

01

Latar Belakang

02

Preprocessing

03

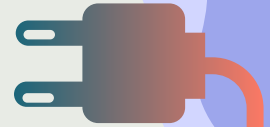
EDA

04

Pemodelan

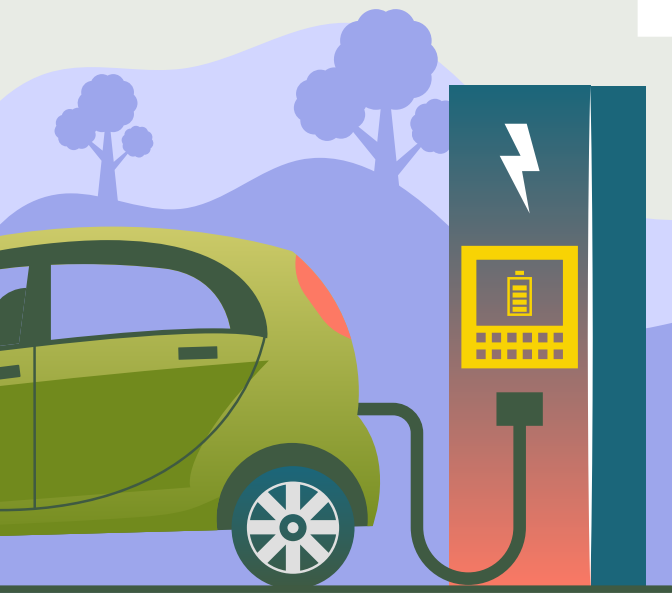
05

Kesimpulan dan Saran



01

Latar Belakang



Di zaman *modern* sekarang ini, teknologi semakin cepat mengalami perkembangan. Munculnya kendaraan listrik menambah “pilihan” lain masyarakat dalam memilih jenis kendaraan yang akan dibeli/digunakan. Oleh karena itu, penting untuk **mengetahui tren dalam penggunaan kendaraan listrik**. Dalam *project* ini, kami akan menggunakan data *Electric Vehicle Title and Registration Activity* untuk **memodelkan jumlah kendaraan listrik di *state* Washington**.



Batasan Masalah

Batasan State

Kami hanya akan
berfokus pada *state*
Washington



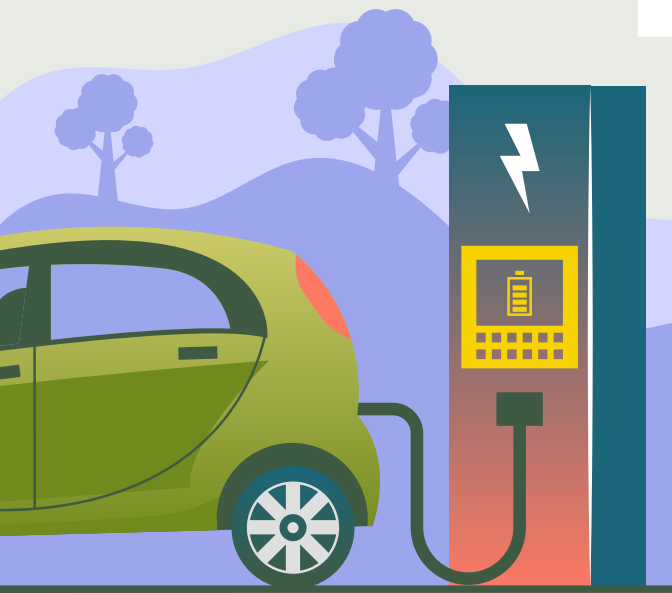
Sumber: https://geology.com/state-map/washington.shtml#google_vignette

Transaction Type

Kami hanya tertarik pada pembelian kendaraan baru dan bukan transaksi kendaraan bekas. Informasi jumlah mobil di jalan dari waktu ke waktu didapatkan dengan menjumlahkan transaksi "Original Title" bulanan untuk setiap *county* secara kumulatif.

02

Preprocessing



Drop Kolom



Drop Kolom yang Tidak Digunakan



```
drop_column = ['2015 HB 2778 Exemption Eligibility', '2019 HB 2042 Clean Alternative Fuel Vehicle (CAFV) Eligibility', 'Meets 2019 HB 2042 Electric Range Requirement', 'Meets 2019 HB 2042 Sale Date Requirement', 'Meets 2019 HB 2042 Sale Price/Value Requirement', '2019 HB 2042: Battery Range Requirement', '2019 HB 2042: Purchase Date Requirement', '2019 HB 2042: Sale Price/Value Requirement', 'Electric Vehicle Fee Paid', 'Transportation Electrification Fee Paid', 'Hybrid Vehicle Electrification Fee Paid', '2020 Census Tract', 'Legislative District', 'Electric Utility']  
df.drop(drop_column, axis = 1, inplace = True)
```



Drop Kolom dengan Informasi Redundan



```
drop_column2 = ['Sale Date', 'Transaction Year', 'Base MSRP']  
df.drop(drop_column2, axis = 1, inplace = True)
```

Proses Selanjutnya



Filter Data

Kami akan fokus pada *title transaction* untuk menghitung jumlah kendaraan listrik yang ada di jalan



Membuat Kolom *Month/Year*



```
df['m/y'] = df['DOL Transaction Date'].dt.strftime("%m-%Y")
```



Mengubah Index Dataframe

Mengubah 'DOL Transaction Date' ke format *datetime* dan menjadikannya sebagai *index*



Data Duplikat



Persis Sama

Tidak ada data duplikat yang persis sama



m/y, DOL Vehicle ID, County

138 baris data duplikat



```
df.drop_duplicates(subset = ['m/y', 'DOL Vehicle ID', 'County'], keep = 'last', inplace = True)
```



m/y, DOL Vehicle ID

84 baris data duplikat



```
df.drop_duplicates(subset = ['m/y', 'DOL Vehicle ID'], keep = 'last', inplace = True)
```

Missing Values



Selain kolom yang ada berikut, tidak memiliki *missing value*.

County	9
City	21
State of Residence	1
Postal Code	3

Analisis kami akan **difokuskan** pada **negara bagian (state) Washington**. Oleh karena itu, kami tidak akan menyimpan kendaraan dengan pemilik yang tinggal di luar WA (Washington).



```
df.dropna(subset = ['County'], inplace = True)
```



```
df = df[df['State of Residence']=='WA']
```



```
df['City'].fillna('Unknown', inplace = True)
```



```
df['Postal Code'].fillna('Unknown', inplace = True)
```

Standarisasi Nama pada Kolom Model



Sebelum

Terdapat 132 nama model yang unik



Standardisasi



```
df['Model'] = df['Model'].map(lambda model: model.title())
df['Model'].unique()
# Kemudian
df['Model'].replace('Prius Plug-In', 'Prius', inplace = True)
```

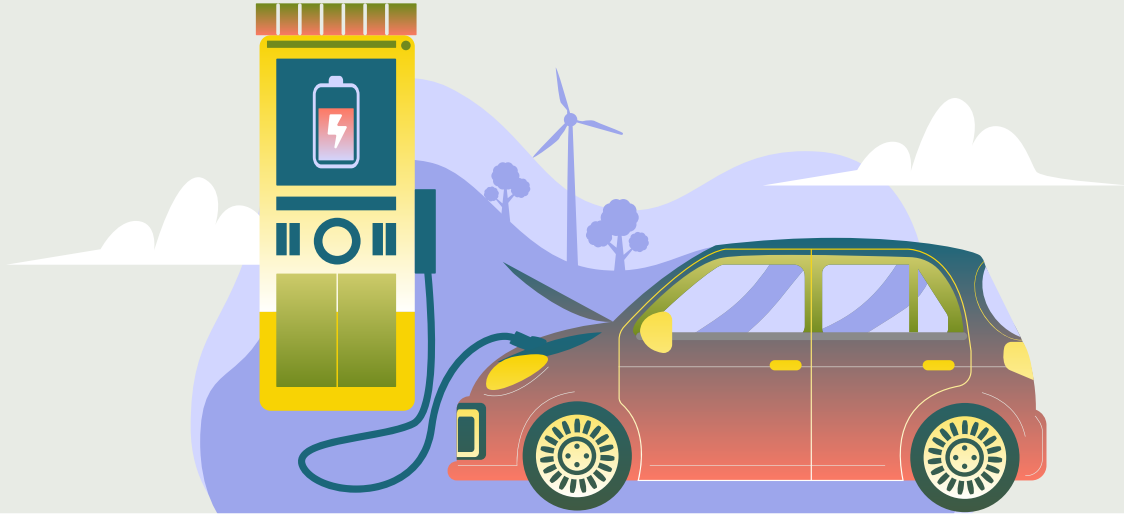


Sesudah

Terdapat 131 nama model yang unik



03



EDA

Exploratory Data Analysis

Filter berdasarkan County



Filter Original Title Transaction

```
1 county_dict = {}  
2 for county in list(df['County'].unique()):  
3     county_dict[county] = df[(df['County']==county) & (df['Transaction  
    Type']=='Original Title')].resample('M').size().cumsum()  
4 # resample('M'): month
```



Bentuk dataframe

```
1 df_cumsum = pd.DataFrame(county_dict)  
2 df_cumsum.fillna(0, inplace = True)  
3 df_cumsum = df_cumsum.reset_index()
```

Jumlah Kendaraan Listrik

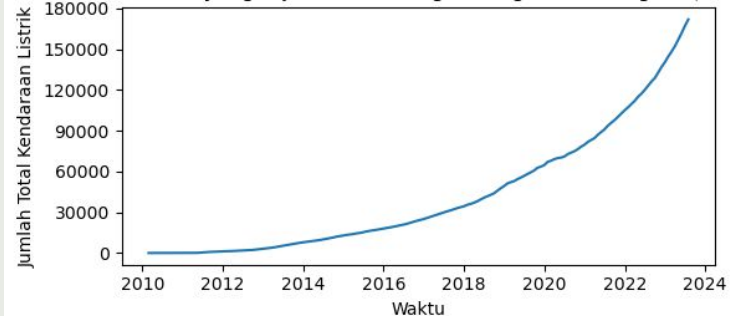
Pada **Negara Bagian Washington**

ada **180 Ribu** kendaraan di tahun 2023



```
1 fig, ax = plt.subplots(figsize = (6, 3))
2 sns.lineplot(x = 'DOL Transaction Date', y = 'State Total', data = df_cumsum, ax = ax)
3 ax.set_xlabel('Waktu')
4 ax.set_ylabel('Jumlah Total Kendaraan Listrik')
5 ax.set_title('Kendaraan Listrik yang Dijalankan di Negara Bagian Washington (2010-2023)')
6 ax.set_yticks(range(0, 200000, 30000))
7 plt.tight_layout()
```

Kendaraan Listrik yang Dijalankan di Negara Bagian Washington (2010-2023)



Jumlah kendaraan listrik meningkat secara eksponensial selama 3 tahun terakhir.

Filter Top 10 County EV Terbanyak

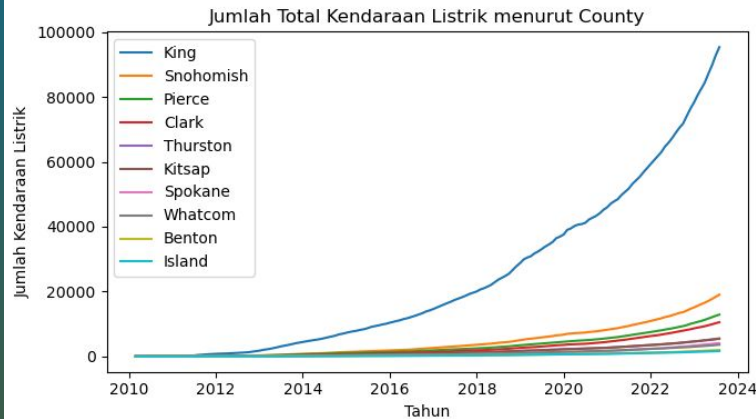
```
1 df['County'].value_counts()
```

King	113658
Snohomish	24021
Pierce	16988
Clark	12063
Thurston	7112
...	...



```
1 top_10_county = ['King', 'Snohomish', 'Pierce', 'Clark', 'Thurston',  
2                 'Kitsap', 'Spokane', 'Whatcom', 'Benton', 'Island']  
3 # Mem-filter top 10  
4 df_cumsum = df_cumsum.loc[:, ['DOL Transaction Date', *top_10_county]]  
5 df_cumsum.head()
```

Jumlah EV berdasarkan County

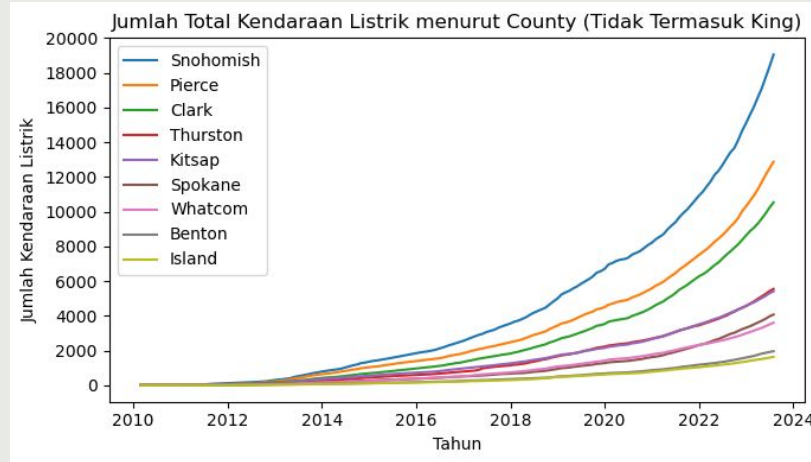


```
1 fig, ax = plt.subplots(figsize = (7, 4))
2 for county in top_10_county:
3     sns.lineplot(x = 'DOL Transaction Date', y = county,
4                 data = df_cumsum, ax = ax, label = county)
5 ax.set_xlabel('Tahun')
6 ax.set_ylabel('Jumlah Kendaraan Listrik')
7 ax.set_title('Jumlah Total Kendaraan Listrik menurut County')
8 ax.legend()
9 plt.tight_layout()
```

County King meningkat
jauh lebih cepat
dibandingkan County
lainnya.

*Berdasarkan data sensus 2020, populasi pada King County ada sebanyak 2.3 juta jiwa.

Jumlah EV berdasarkan County



Snohomish

Jumlah kendaraan terbanyak kedua.



```
1 fig, ax = plt.subplots(figsize = (7, 4))
2 for county in top_10_county:
3     sns.lineplot(x = 'DOL Transaction Date', y = county,
4                 data = df_cumsum, ax = ax, label = county)
5 ax.set_xlabel('Tahun')
6 ax.set_ylabel('Jumlah Kendaraan Listrik')
7 ax.set_title('Jumlah Total Kendaraan Listrik menurut County')
8 ax.legend()
9 plt.tight_layout()
```

Pierce & Clark

Kemudian diikuti urutan ketiga dan keempat

Jumlah EV berdasarkan Model

Tesla Model 3

Kendaraan terbanyak pertama

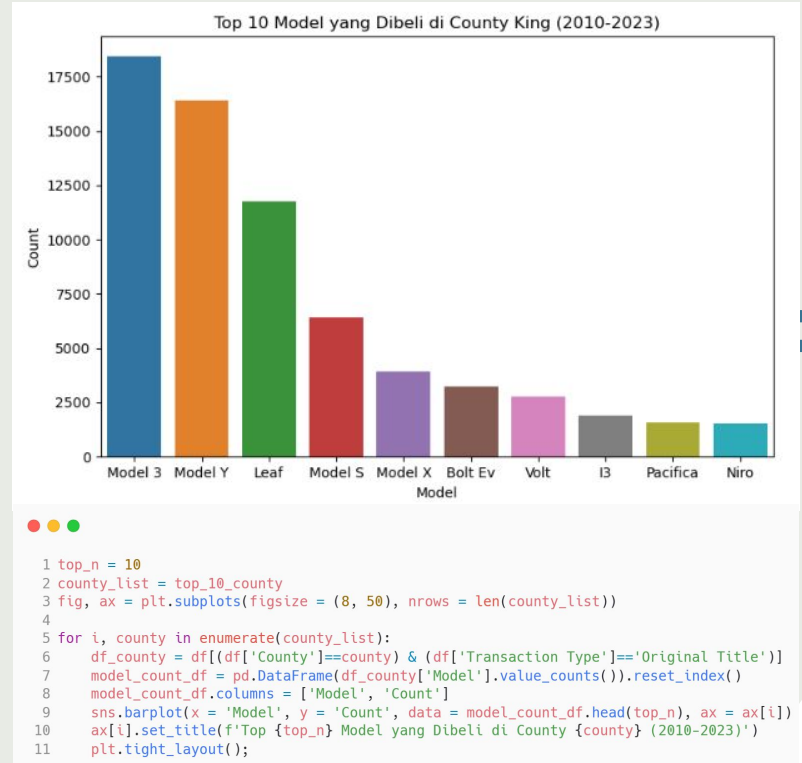
Tesla Model Y

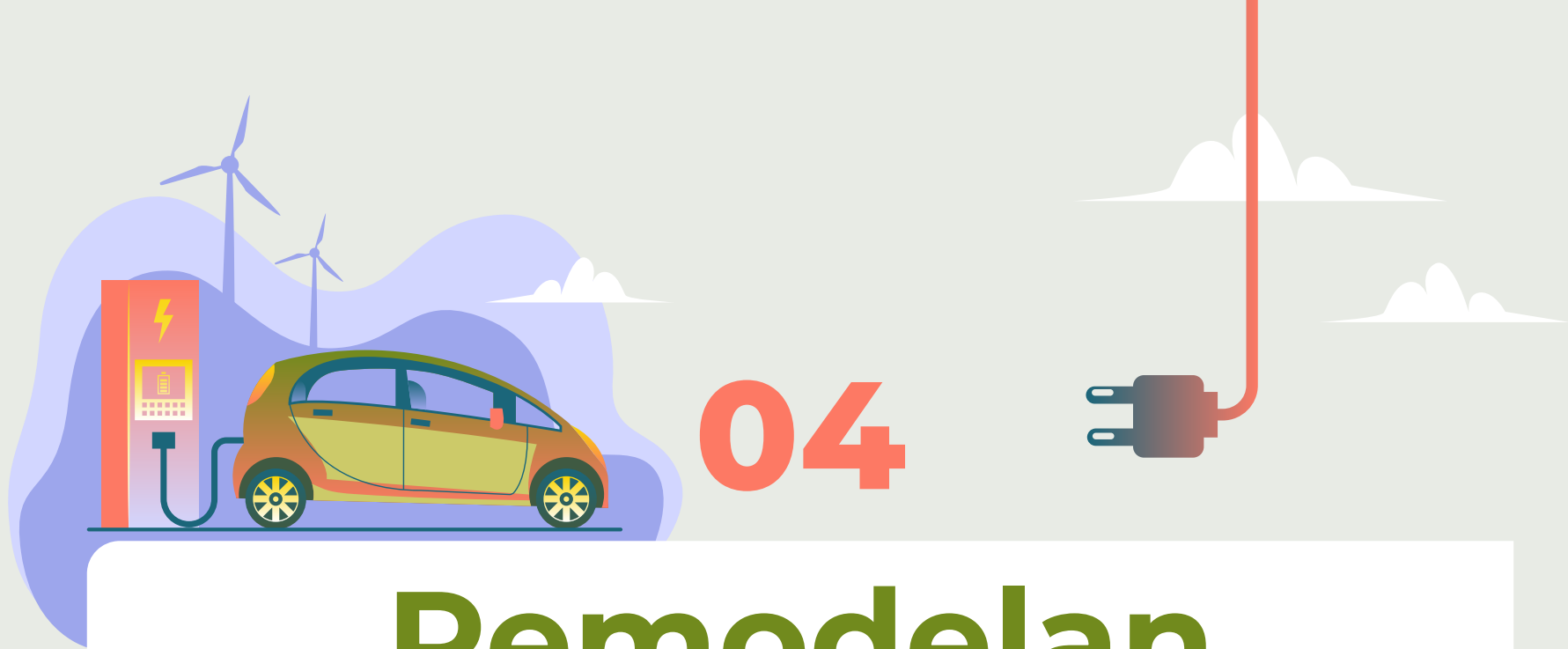
Kendaraan terbanyak kedua

Nissan Leaf

Kendaraan terbanyak ketiga

Kendaraan di atas adalah top 3 kendaraan listrik yang paling banyak digunakan pada setiap county.

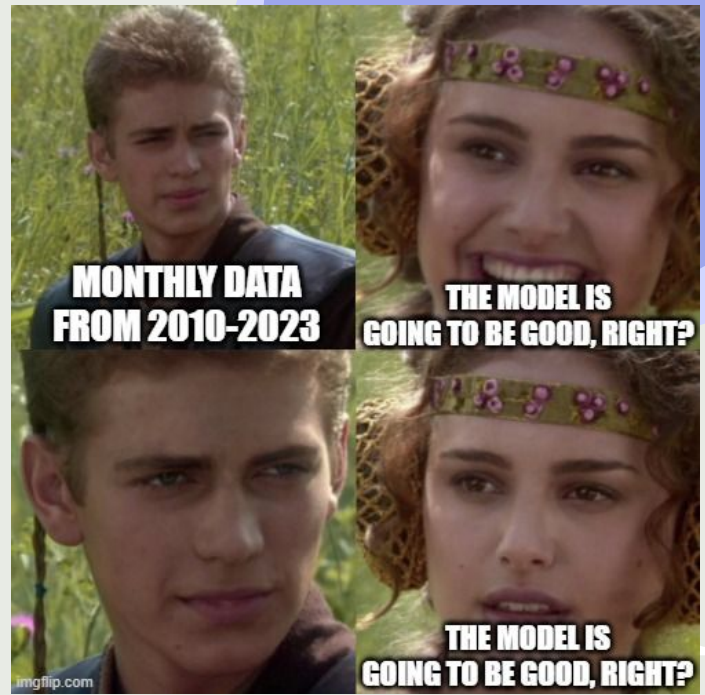




Pemodelan

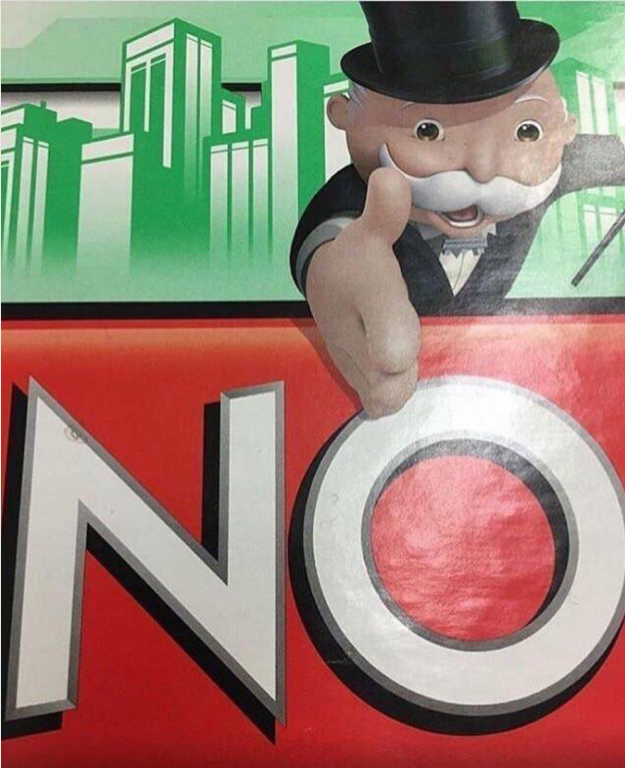


Semakin banyak data yang digunakan
(diasumsikan sudah di-*preprocessing*)
untuk melatih model, maka model akan
semakin baik



Yes / No

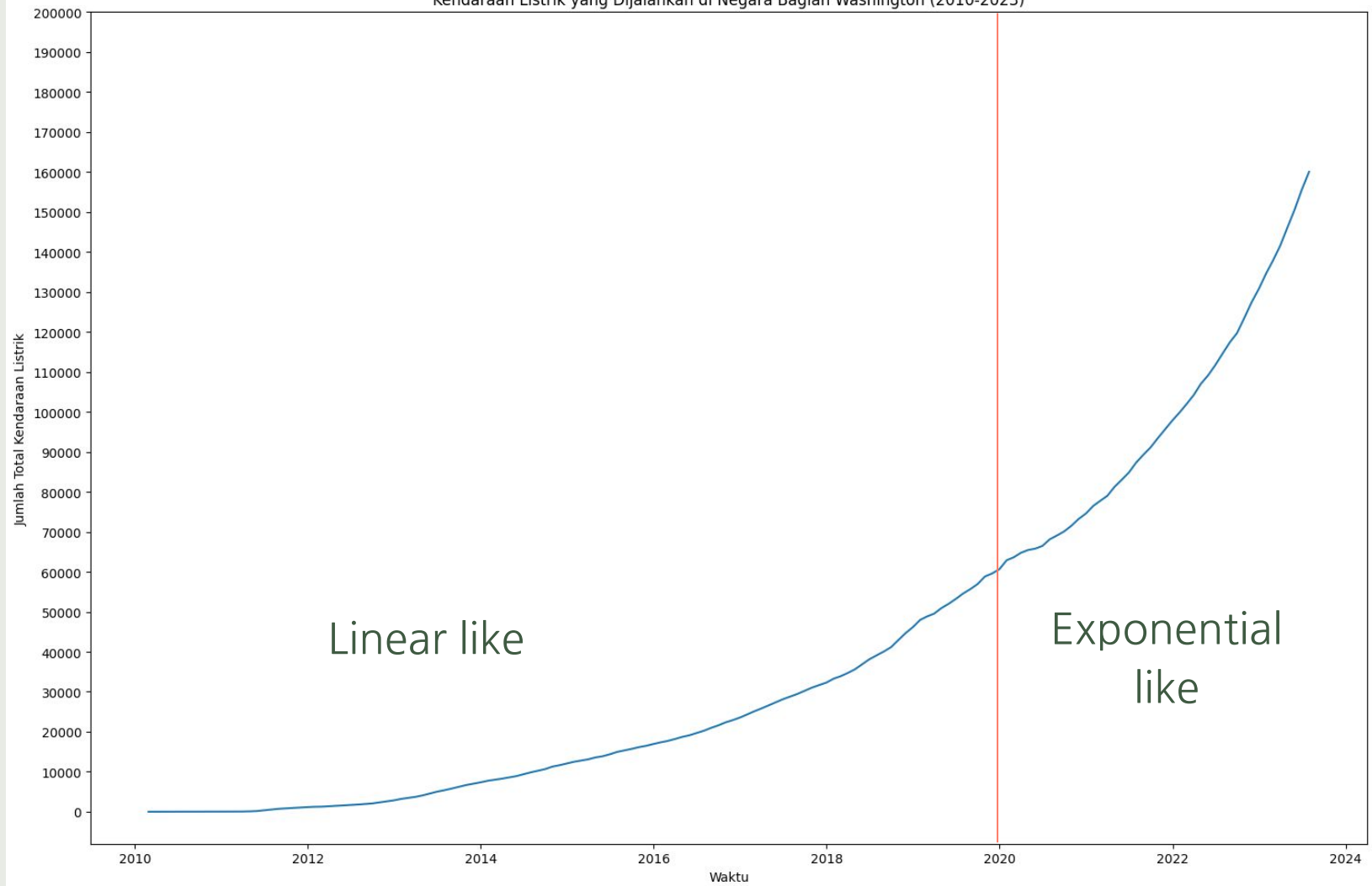
No



Tell Me Why

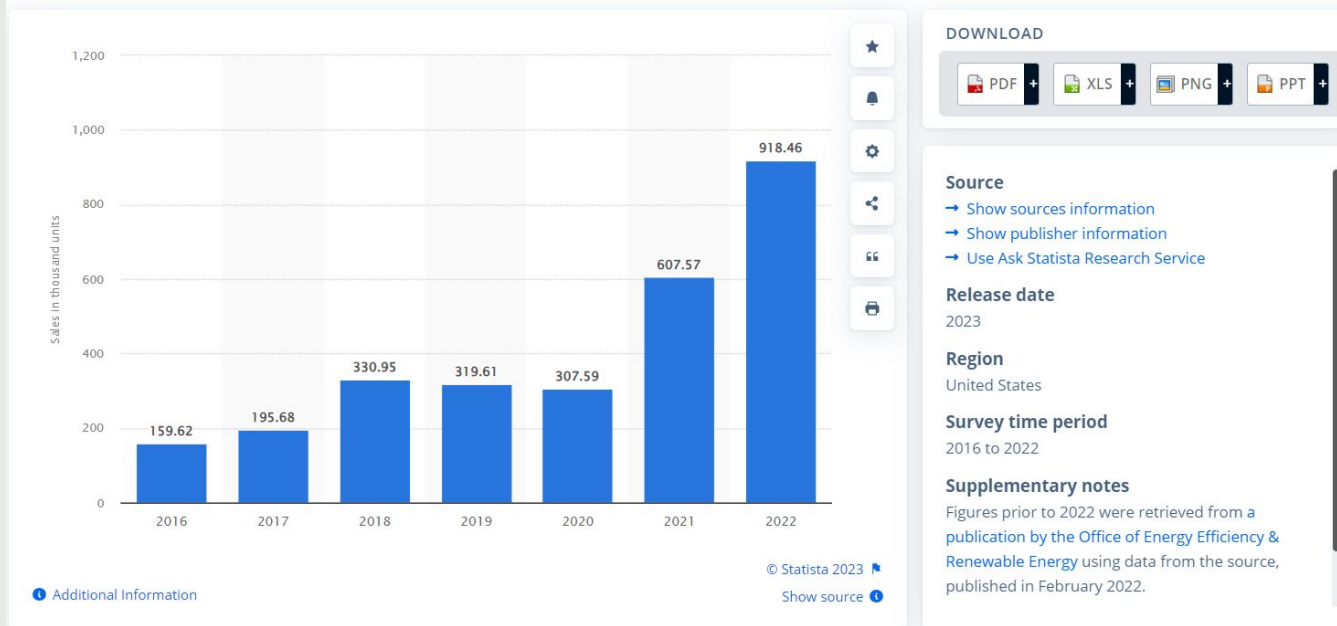


Kendaraan Listrik yang Dijalankan di Negara Bagian Washington (2010-2023)



Estimated plug-in electric light vehicle sales in the United States from 2016 to 2022

(in 1,000 units)



Penjualan EV mulai mengalami peningkatan yang drastis dimulai dari 2020

Sumber: <https://www.statista.com/statistics/665823/sales-of-plug-in-light-vehicles-in-the-us/>

Data yang akan digunakan untuk melatih model akan dimulai dari 2020

Langkah Pemilihan Model

Step 1

Gunakan data 2010-2023,
lakukan *feature engineering* &
hyperparameter tuning

Step 2

Plot model yang diperoleh
dari *Step 1* dengan data
test, perhatikan nilai
metrics yang diperoleh

Step 3

Lakukan *Step 1* & 2 dengan
data 2020-2023, kemudian
bandingkan nilai *metrics*
yang diperoleh dengan
model (2010-2023)

Feature Engineering & Hyperparameter Tuning




```
def evaluate_model(model):  
    display(model.summary())  
    model.plot_diagnostics()  
    plt.tight_layout()
```

- Membuat custom *function* untuk mengecek signifikansi parameter, nilai AIC, BIC, dll.
- Digunakan `auto.arima()` untuk menentukan orde p, d, q dan P, D, Q terbaik



```
auto_model = pm.auto_arima(train_king, start_p = 0, start_d = 0, start_q = 0, max_p = 4,  
                           max_d = 3, max_q = 4, start_P = 0, start_D = 0, start_Q = 0, max_P = 3,  
                           max_D = 3, max_Q = 3, m = 12)  
auto_model.summary()
```

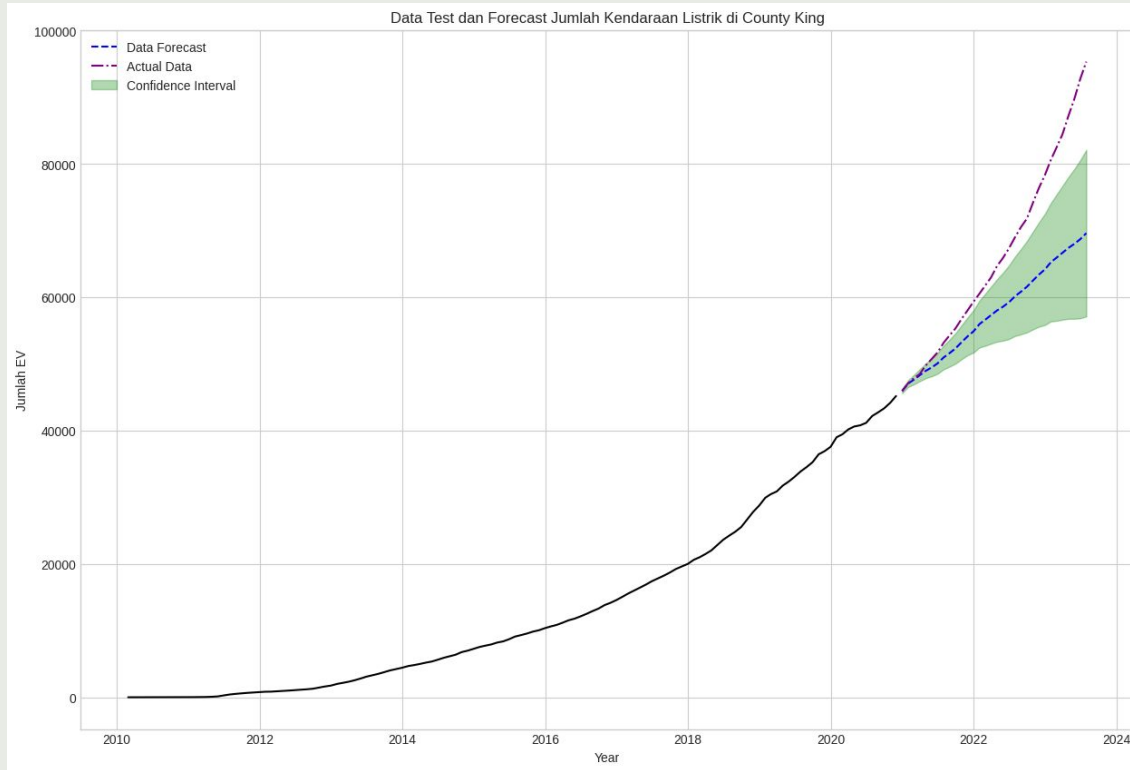
Feature Engineering & Hyperparameter Tuning



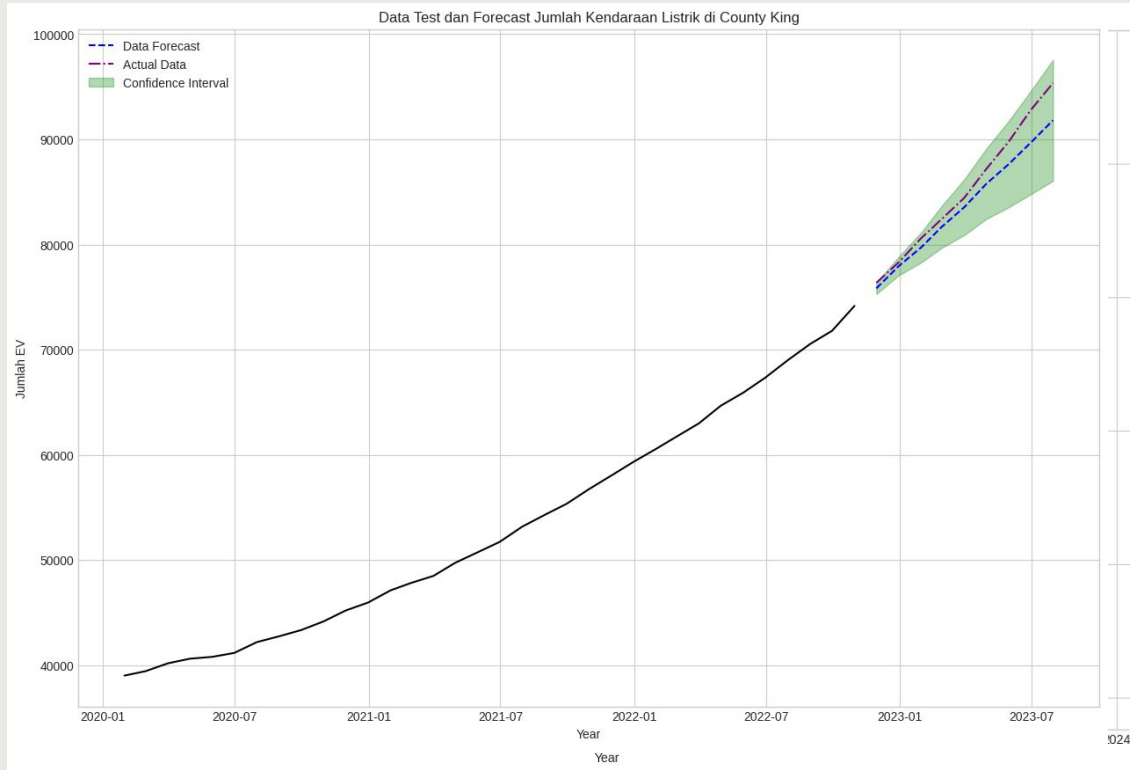
```
model = SARIMAX(train_king, order = (p, d, q), seasonal_order = (P, D, Q, 12),  
                enforce_invertibility = False, enforce_stationarity = False).fit()  
evaluate_model(model)
```

- Masukkan orde yang telah dihasilkan di `auto.arima()` itu ke dalam variabel “model” untuk membuat model SARIMAX secara manual

County King (2010-2023)



County King (2020-2023)



Mean Absolute Percentage Error

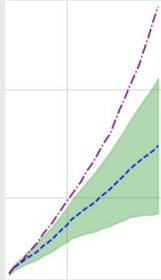
<i>MAPE</i>	Forecasting power
<10%	Highly accurate forecasting
10%~20%	Good forecasting
20%~50%	Reasonable forecasting
>50%	Weak and inaccurate forecasting

Source: Lewis (1982)

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

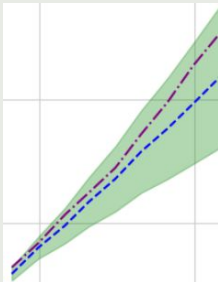
Perbandingan Model County King

2010-2023



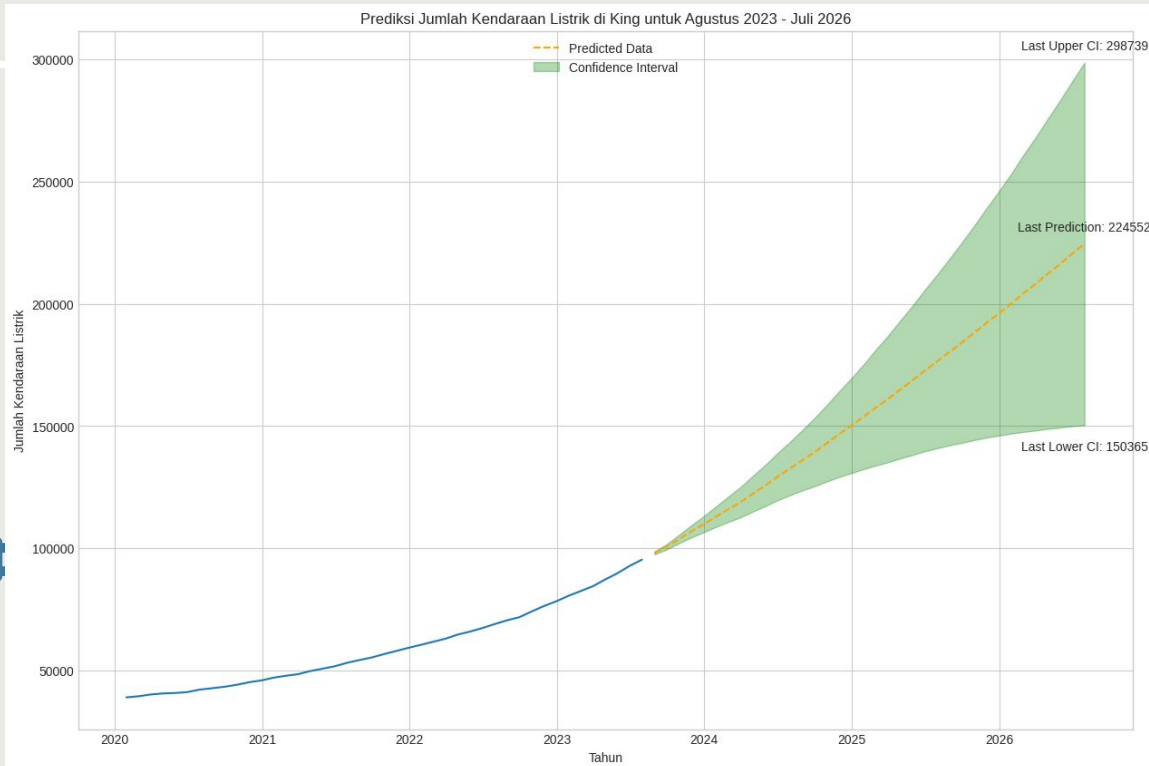
King	2010-2023	2020-2023
Mean Absolute Percentage Error	11%	2%
Mean Squared Error	125.719.126	3.470.400
Root Mean Squared Error	11.212	1863
R-Squared	0,39	0,91

2020-2023



- *Value* pada data *test* masuk ke dalam *Confidence Interval* (CI) model 2020-2023 vs tidak masuk ke CI model 2010-2023
- **Metrics** pengukuran lebih baik pada model 2020-2023
- Kesimpulan = model 2020-2023 lebih baik

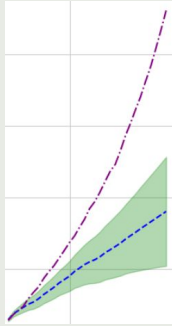
Prediksi Jumlah EV County King



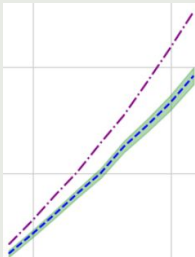
Pada Juli 2026,
diprediksi akan ada
224.552 jumlah EV di
County King

Perbandingan Model County Snohomish

2010-2023



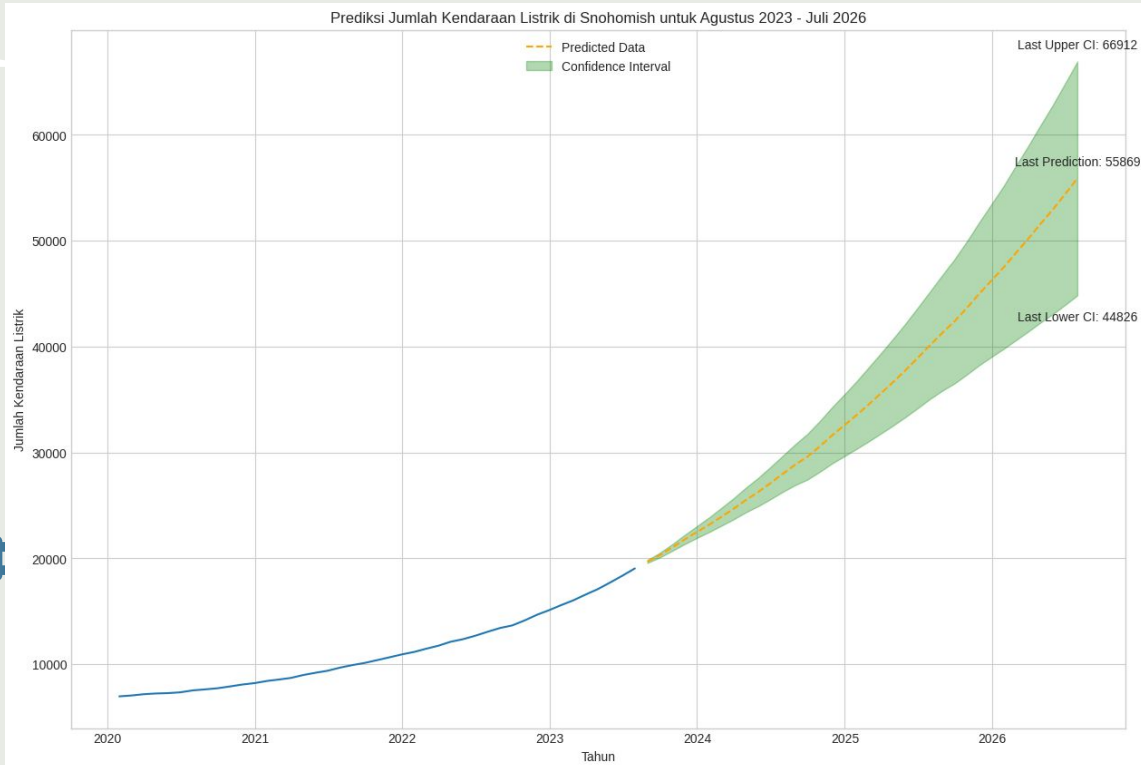
2020-2023



Snohomish	2010-2023	2020-2023
Mean Absolute Percentage Error	15%	3%
Mean Squared Error	9.548.501	470.624
Root Mean Squared Error	3.090	686
R-Squared	0,02	0,76

- *Value* pada data *test* lebih berdekatan ke *Confidence Interval* (CI) model 2020-2023 dibanding CI model 2010-2023
- **Metrics** pengukuran lebih baik pada model 2020-2023
- Kesimpulan = model 2020-2023 lebih baik

Prediksi Jumlah EV County Snohomish



Pada Juli 2026,
diprediksi akan ada
55.869 jumlah EV di
County Snohomish

Perbandingan Model-Model Untuk County Lainnya

Pierce	2010-2023	2020-2023
Mean Absolute Percentage Error	13%	4%
Mean Squared Error	3.035.718	255.836
Root Mean Squared Error	1.742	506
R-Squared	0,32	0,71

Clark	2010-2023	2020-2023
Mean Absolute Percentage Error	17%	2%
Mean Squared Error	2.794.525	78.695
Root Mean Squared Error	1,672	281
R-Squared	0,10	0,84

Perbandingan Model-Model Untuk County Lainnya

Thurston	2010-2023	2020-2023
Mean Absolute Percentage Error	14%	1%
Mean Squared Error	609.168	6.118
Root Mean Squared Error	780	78
R-Squared	0,12	0,95

Kitsap	2010-2023	2020-2023
Mean Absolute Percentage Error	10%	1%
Mean Squared Error	357.052	12.295
Root Mean Squared Error	598	111
R-Squared	0,45	0,87

Perbandingan Model-Model Untuk County Lainnya

Spokane	2010-2023	2020-2023	Whatcom	2010-2023	2020-2023
Mean Absolute Percentage Error	17%	1%	Mean Absolute Percentage Error	10%	3%
Mean Squared Error	460.540	4.482	Mean Squared Error	152.136	14.860
Root Mean Squared Error	679	67	Root Mean Squared Error	390	122
R-Squared	0,14	0,94	R-Squared	0,45	0,69

Perbandingan Model-Model Untuk County Lainnya

Benton	2010-2023	2020-2023
Mean Absolute Percentage Error	13%	5%
Mean Squared Error	68.820	14,641
Root Mean Squared Error	262	121
R-Squared	0,34	0,21

Island	2010-2023	2020-2023
Mean Absolute Percentage Error	13%	2%
Mean Squared Error	40.528	1,194
Root Mean Squared Error	201	35
R-Squared	0,34	0,87

Prediksi Jumlah EV Untuk Juli 2026 di Semua County

Nama County	Prediksi
King	224.552
Snohomish	55.869
Pierce	26.924
Clark	21.544
Thurston	10.660

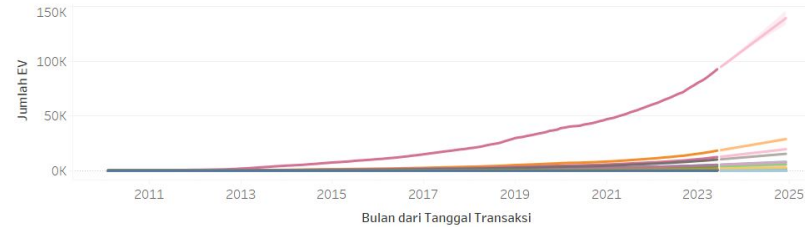
Nama County	Prediksi
Kitsap	9.869
Spokane	8.193
Whatcom	7.169
Benton	3.516
Island	3.098

Dashboard

Proyeksi Jumlah Kendaraan Listrik di Negara Bagian Washington

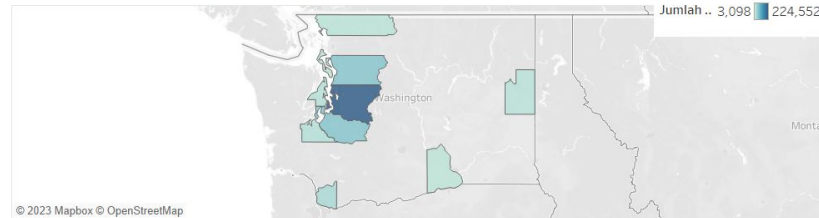
Pilih Negara Bag (All)

Jumlah EV Per County Forecast



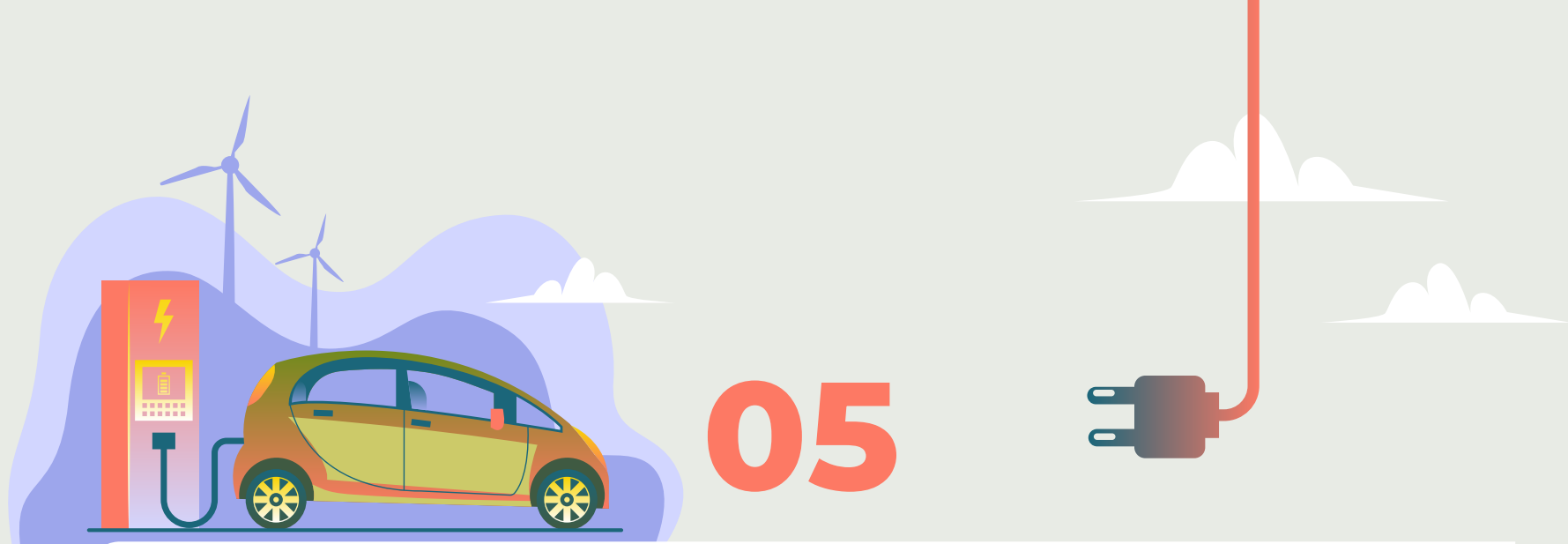
*Nilai setelah Juni 2023 adalah prediksi.

Proyeksi Jumlah EV (Juli 2026)



*Proyeksi hanya mencakup 10 wilayah (Benton, Clark, Island, King, Kitsap, Pierce, Snohomish, Spokane, Thurston, Whatcom). Jika tidak s...





Kesimpulan dan Saran



Kesimpulan dan Saran



Kesimpulan

- Tren EV meningkat secara eksponensial selama 3 tahun terakhir.
- Kendaraan EV paling diminati adalah Tesla, Nissan Leaf, dan Chevy Volt.
- Forecasting yang dilakukan sudah baik karena $MAPE < 10\%$.



Saran

- Memperbanyak *charging station* di *county-county* dengan jumlah EV yang banyak.
- Menyertakan adaptor untuk Tesla, Nissan Leaf, dan Chevy Volt di *charging station* baru.



THANKS!

