




Stochastic *Approach* in Exponential Decline Curve Analysis



Anggota



Alfian Isnan

1806148643



**Hanvey
Xavero**

1806200135



**Muhammad
As'ad Muyassir**

1806199953



**Vincentius
Indra**

1806200362



Data and Source

Laju Produksi Panas Bumi



Paper source : <https://fmipa.unmul.ac.id/files/docs/1.%20Sri.pdf>

Tabel Laju Produksi Panas Bumi
(ton/jam) pada Tahun 1996

Bulan	Laju Produksi	<i>Loss</i>	<i>Loss Ratio</i>
Januari	77,234	-0,251	-308,005
Februari	75,313	-1,921	-39,212
Maret	74,296	-1,017	-73,044
April	72,600	-1,696	-42,812
Mei	72,133	-0,467	-154,490
Juni	71,135	-0,999	-71,226
Juli	70,351	-0,783	-89,810
Agustus	70,262	-0,089	-788,558
September	69,610	-0,652	-106,699
Oktober	69,263	-0,347	-199,612



Data and Source

Data yang digunakan adalah data laju produksi panas bumi yang diambil dari suatu sumur geothermal di Kamojang, Jawa Barat, tahun 1996



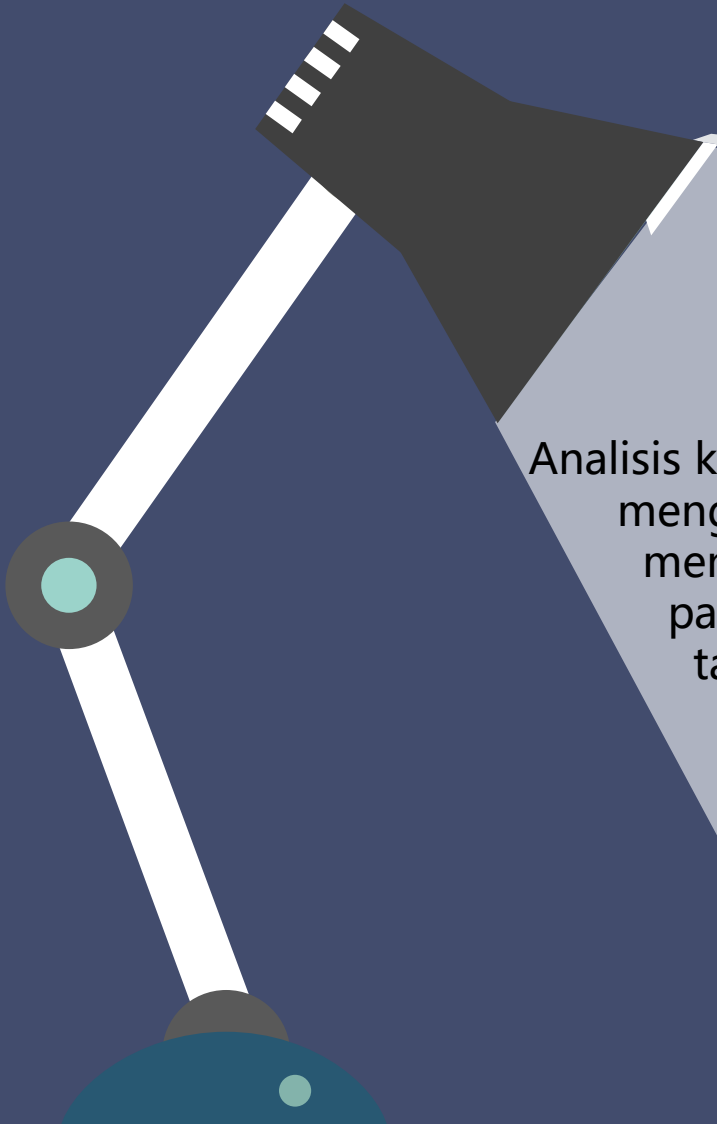
Metode

Arps Eksponential Method



Metode

Arps Eksponential Method



Analisis kurva penurunan menggunakan typecurves untuk menganalisis data produksi. Sri Wahyuningsih menegaskan bahwa pembuatan tulisan yang berjudul Pendekatan Stokastik pada Analisa Kurva Penurunan Tipe Eksponensial ini adalah untuk melakukan tantangan investigasi pada estimasi cadangan dengan metode probabilitas pada analisis kurva penurunan.

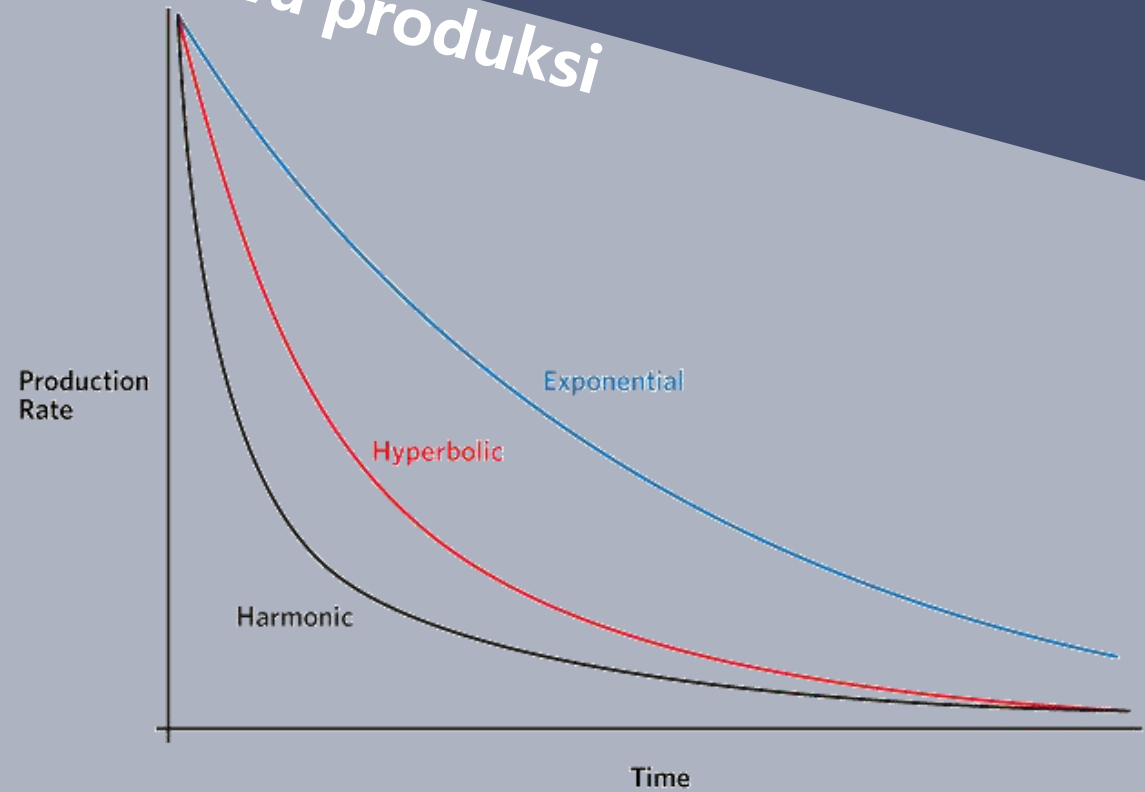
Kemudian untuk membangun representasi dari penggunaan persamaan Arps dengan pendekatan stokastik. Dengan didefinisikannya persamaan Arps sebagai berikut :

$$q_t = \frac{q_0}{(1 + bDt)^{1/b}}$$

Metode

Arps Eksponential Method

Jenis-jenis kurva produksi



Metode

Arps Eksponential Method

Fungsi

Arps Equation

$$q_t = \frac{q_i}{(1 + bD_i t)^{1/b}}$$

Detail :

b = Decline Exponent

q_i = Initial Rate, Mscf/day

t = Time, days

D_i = Initial decline rate, day⁻¹

Exponential	$b = 0$	$q_t = q_i \exp(-D_i t)$
Hyperbolic	$0 < b < 1$	$q_t = \frac{q_0}{(1 + bD_i t)^{1/b}}$
Harmonic	$b = 1$	$q_t = \frac{q_i}{(1 + D_i t)}$

Metode

Arps Eksponential Method

Penurunan tipe hiperbolik, yang pada kenyataannya paling banyak diaplikasikan dapat dicocokkan dengan fakta bahwa *loss ratio* mendekati deret aritmatika, serta turunan pertama dari *loss ratio* mendekati konstan, yang dapat diterangkan dengan persamaan diferensial sebagai berikut :

$$\frac{d\left(\frac{q_t}{dq_t / dt}\right)}{dt} = b,$$

dimana $b < 0$.



Metode

Arima (1, 1, 0) Method



Metode

ARIMA (1, 1, 0) Method



Laju Produksi q_t pada interval waktu yang sama ditabulasikan dalam satu kolom, penurunan produksi per unit waktu dinotasikan dengan delta q , dan ratio keduanya ditempatkan pada kolom ke-3. Kurva laju-waktu untuk kasus penurunan eksponensial mempunyai lost ratio yang konstan. Persamaan dapat ditulis sebagai

$$\frac{q_t}{\Delta q_t / \Delta t} = a$$

atau dapat juga ditulis sebagai

$$q_t = \frac{-a}{\Delta t - a} q_{t-1}$$

Karena ada tren menurun pada pola ini, maka model ini adalah model autoregresi setelah dilakukan pembedaan (differencing) pertama, yang merupakan model ARIMA (1,1,0). (Wei, 1990).



Result

Arima (1, 1, 0) vs Arps Eksponensial



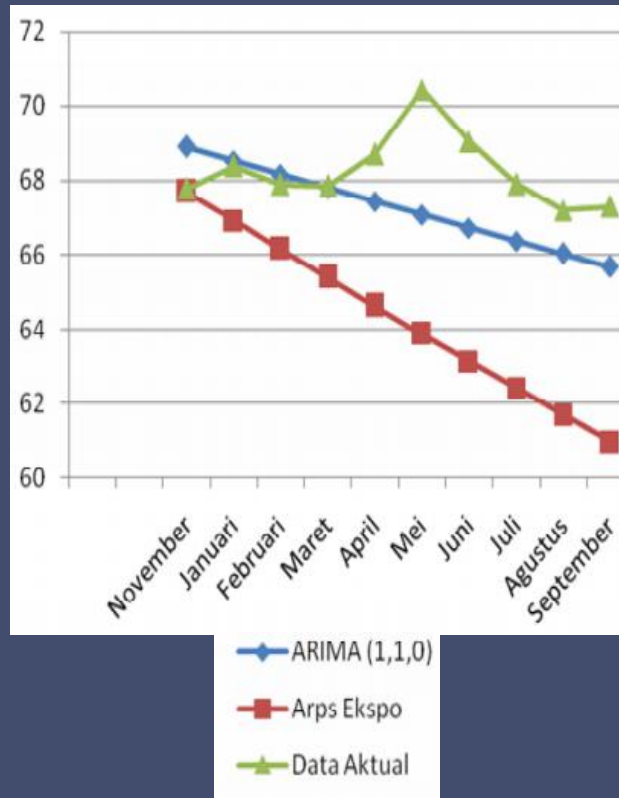
Result

Prediksi laju produksi geothermal

Tabel Prakiraan Laju Produksi Panas Bumi (ton/jam) November 1996 - September 1997

Bulan	ARIMA (1, 1, 0)	Arps Eksponsensial	Data Aktual
November	68,895	67,704	67,781
Januari	68,529	66,921	68,385
Februari	68,165	66,147	67,902
Maret	67,803	65,382	67,858
April	67,443	64,626	68,705
Mei	67,085	63,879	70,434
Juni	66,729	63,140	69,042
Juli	66,375	62,410	67,916
Agustus	66,022	61,689	67,201
September	65,672	60,975	67,307

Grafik perbandingan Laju Produksi Panas Bumi (ton/jam)



MSE (Mean Square Error)

- ARIMA (1, 1, 0)
 - MSE = 10,159
- Arps Eksponsensial
 - MSE = 20,659
- Pada studi kasus tersebut, pendekatan menggunakan model ARIMA (1, 1, 0) lebih mendekati data aktual dibandingkan dengan metode Arps eksponensial.



Re-Implementation

Pengguna Kereta Jabodetabek





P r e d i k s i Jumlah Penumpang KRL 2018-2021



Re-Implementation

Kelompok kami ingin melakukan re-implementasi kedua metode tersebut ke data pelanggan kereta Jabodetabek dari PTKA tahun 2006-2019.

Kami akan mencoba menggunakan data tahun 2014 - 2017, untuk memprediksi jumlah pelanggan pada tahun 2018 - 2021. dengan 2 metode yang telah dijelaskan sebelumnya.

Dan membandingkan data hasil prediksi dengan data aktual.



Re-Implementation

Method and Code



**Arps
Eksponensial**



Arima (1, 1, 0)

**Click picture to open website*

Re-Implementation

Method and Code

ARPS Exponential Equation

persamaan ARPS dinyatakan dengan

$$Q_t = Q_i * (1 + bDt)^{\frac{1}{b}}$$

q = current production rate

q i = initial production rate (start of production)

d i = initial nominal decline rate at t = 0

t = cumulative time since start of production

b= hyperbolic decline constant (0 < b < 1)

$$a = \frac{qt}{dq/dt}$$
$$D = \frac{1}{a}$$

b formula is :

$$\frac{d\left(\frac{qt}{(dq/dt)}\right)}{dt}$$

```
In [11]: import math
import pandas
import numpy as np
import matplotlib.pyplot as plt
```

DATA DARI 2014 - 2017 UNTUK PREDIKSI 2018

```
In [12]: #Pengambilan data dari csv
dd = pandas.read_csv('Kereta eR eL.csv', delimiter=';', index_col='Bulan')
nama_bulan= ("Januari", "Februari", "Maret", "April", "Mei", "Juni", "Juli", "Agustus", "September", "Oktober", "November", "Deser")
```

rumus awal

ARPS
EXponential

library yang digunakan

data untuk
prediksi

Re-Implementation

Method and Code

```
In [71]: a=12.192573
b=352.15449
D1 = 1/a
output=[]
for t in range (49, 61):
    hasil = (((b*D1*t)+1)**(1/b))*datad[46]
    output.append(hasil)

In [72]: print("Hasil prediksi pengguna KRL tahun 2018 dari data tahun 2014 - 2017\n")
for i in range(len(output)):
    print(nama_bulan[i], end=': ')
    print(output[i])
```

Hasil prediksi pengguna KRL tahun 2018 dari data tahun 2014 - 2017

Januari: 28834.015778079236
Februari: 28835.668844259268
Maret: 28837.28928870223
April: 28838.87837577953
Mei: 28840.437297717777
Juni: 28841.96717998646
Juli: 28843.4690861918
Agustus: 28844.944022530148
September: 28846.392941847706
Oktober: 28847.816747347515
November: 28849.21629597998
Desember: 28850.592401548736

penerapan
rumus

hasil data
prediksi
2018

Re-Implementation

Method and Code

```
In [73]: mse=[]
        rata_mse=0
        for i in range(len(output)):
            m=(output[i]-dd['Pengguna'][i+48])**2
            mse.append(m)
            rata_mse+=m

In [74]: print("Hasil Mean Square Error setiap bulan\n")
        for i in range(len(output)):
            print(nama_bulan[i], end=': ')
            print(int(mse[i]))
```

Hasil Mean Square Error setiap bulan

Januari: 576104
Februari: 12066375
Maret: 148772
April: 10634
Mei: 23889
Juni: 16071817
Juli: 58821
Agustus: 557925
September: 1508949
Oktober: 220132
November: 640346
Desember: 122785

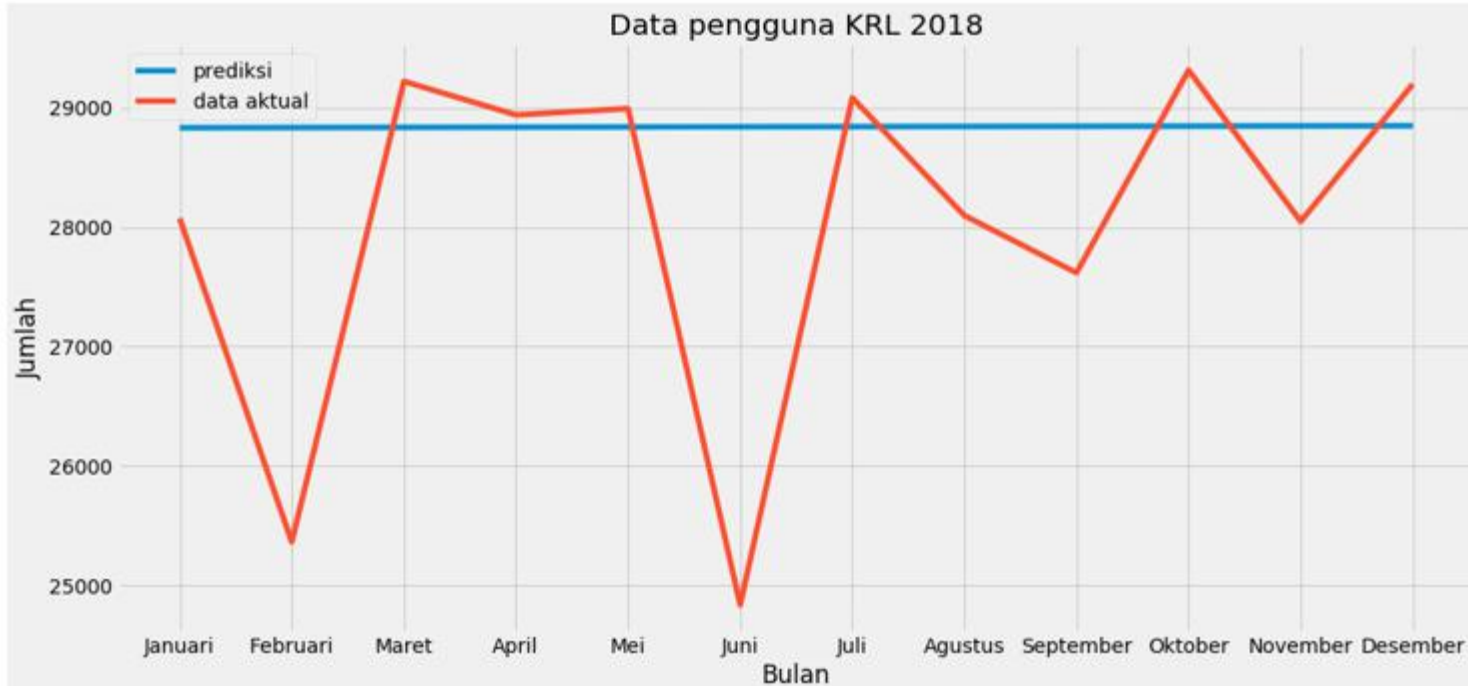
penerapan
rumus mse

mse tiap bulan
(dibandingkan
dengan data
aktual)

Re-Implementation

Method and Code

```
In [76]: df = pandas.read_csv('Kereta RL.csv', delimiter=';', index_col='Bulan')
x = (df['Pengguna'])[11:24]
y = (output)
arps = y
x=x[1:len(x)]
plt.figure(figsize=(15,7))
plt.plot(nama_bulan, y, label='prediksi')
plt.plot(nama_bulan, x, label='data aktual')
plt.xlabel('Bulan')
plt.ylabel('Jumlah')
plt.title('Data pengguna KRL 2018')
plt.grid(True)
plt.legend()
plt.show()
```



code untuk
grafik

grafik Data
pengguna KRL
2018 (prediksi-
data aktual)

Re-Implementation

Method and Code

PREDIKSI MENGGUNAKAN SARIMA

```
In [103]: import warnings
import itertools
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')

data = sm.datasets.co2.load_pandas()
y = data.data

def parser(x):
    return pd.datetime.strptime('20'+x, '%Y-%m')

y = pd.read_csv('dataKRL.csv', header=0, parse_dates=[0], index_col=0, squeeze=True, date_parser=parser)['2014-01-01:']

print(y)
```

library untuk
penggunaan
metode
SARIMA

```
In [104]: # Define the p, d and q parameters to take any value between 0 and 2
p = d = q = range(0, 2)

# Generate all different combinations of p, q and q triplets
pdq = list(itertools.product(p, d, q))

# Generate all different combinations of seasonal p, q and q triplets
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]

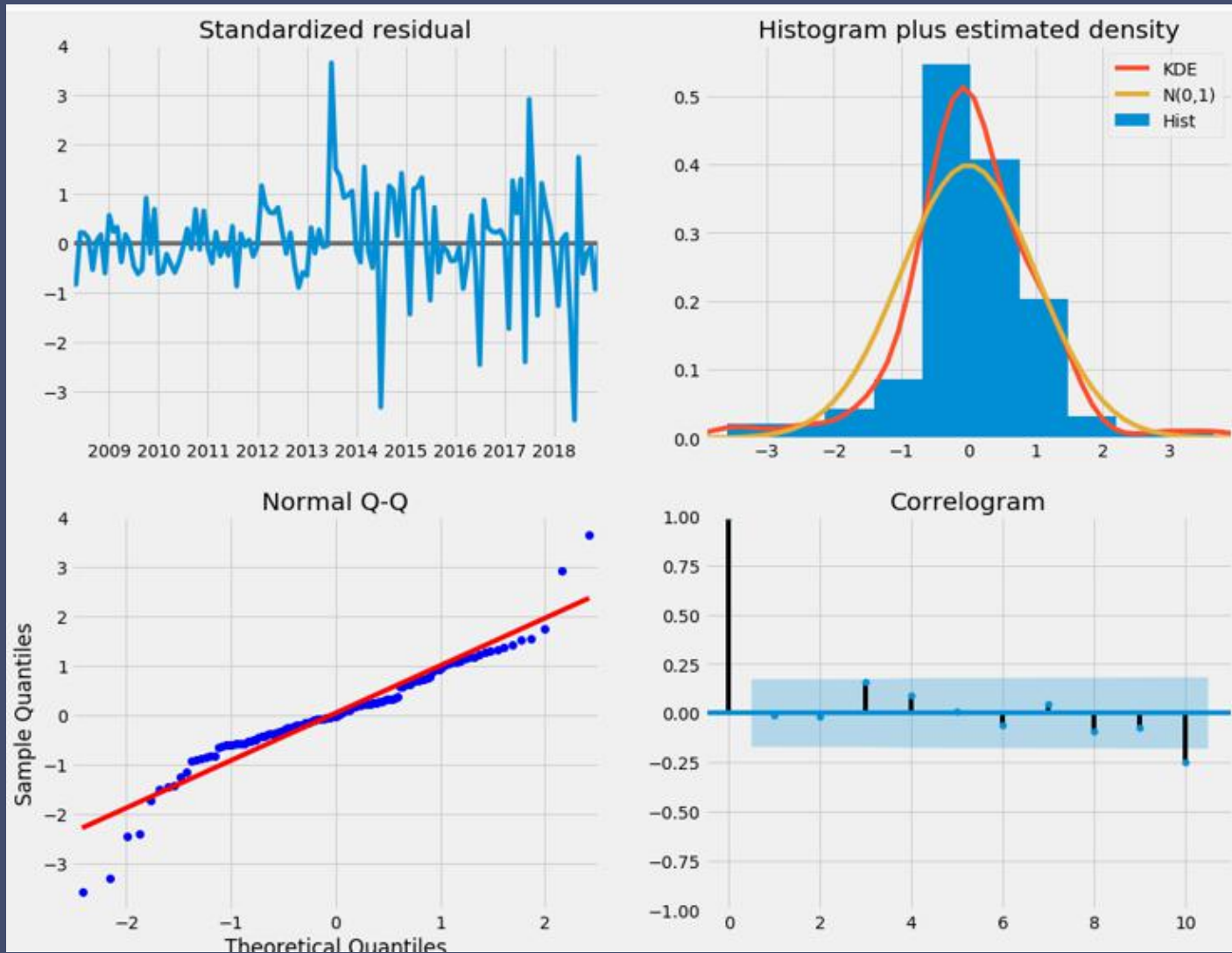
print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))

Examples of parameter combinations for Seasonal ARIMA...
SARIMAX: (0, 0, 1) x (0, 0, 1, 12)
SARIMAX: (0, 0, 1) x (0, 1, 0, 12)
SARIMAX: (0, 1, 0) x (0, 1, 1, 12)
SARIMAX: (0, 1, 0) x (1, 0, 0, 12)
```

Mencari
kombinasi
SARIMAX yang
paling cocok.

Re-Implementation

Method and Code



grafik

Hasil plot
metode
SARIMAX ke
berbagai
macam bentuk
diagram

Re-Implementation

Method and Code

```
In [108]: pred = results.get_prediction(start=pd.to_datetime('2018-01-01'), dynamic=False)
pred_ci = pred.conf_int()
plt.figure(figsize=(15,8))
ax = y['2017:'].plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='One-step ahead Forecast', alpha=.7)

ax.fill_between(pred_ci.index,
               pred_ci.iloc[:, 0],
               pred_ci.iloc[:, 1], color='k', alpha=.2)

ax.set_xlabel('Tahun')
ax.set_ylabel('Penumpang (Ribu)')
plt.legend()

plt.show()
```



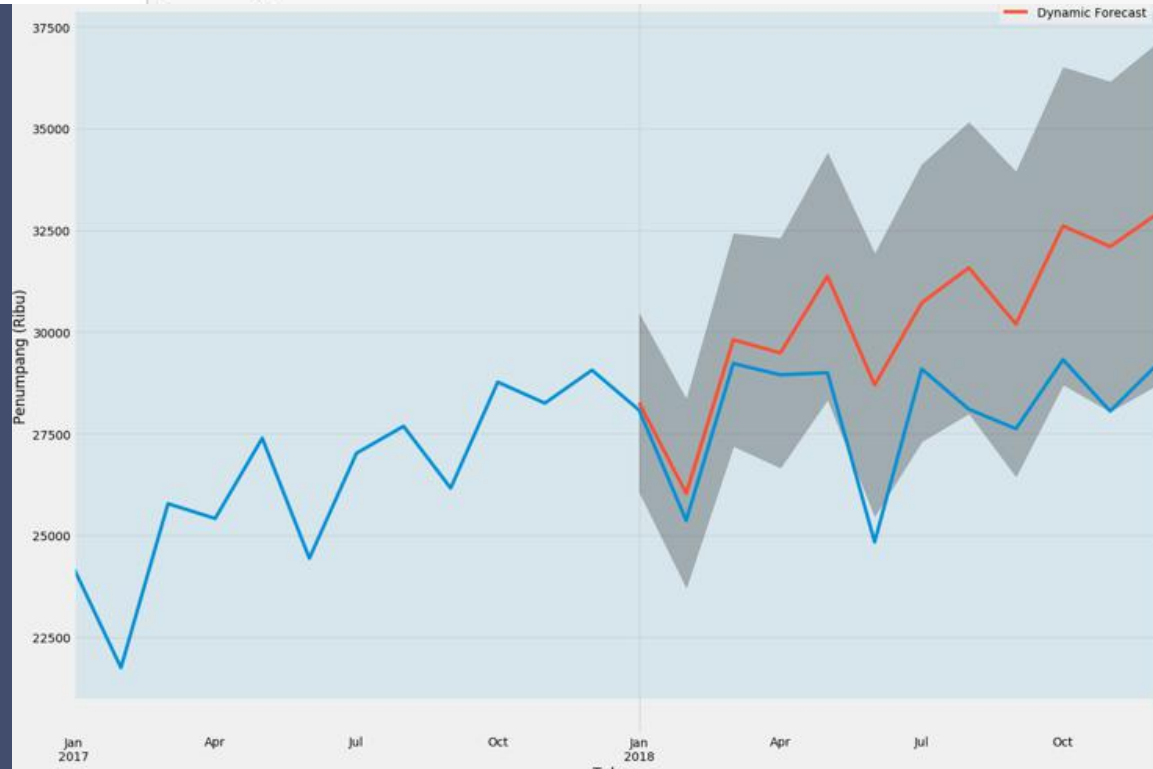
code untuk
plotting hasil
prediksi metode
one step ahead

grafik
perbandingan data
prediksi ARIMA
one step ahead
dengan data aktual
(2018)

Re-Implementation

Method and Code

```
In [110]: pred_dynamic = results.get_prediction(start=pd.to_datetime('2018-01-01'), dynamic=True, full_results=True)
pred_dynamic_ci = pred_dynamic.conf_int()
ax = y['2017:'].plot(label='observed', figsize=(20, 15))
pred_dynamic.predicted_mean.plot(label='Dynamic Forecast', ax=ax)
ax.fill_between(pred_dynamic_ci.index,
               pred_dynamic_ci.iloc[:, 0],
               pred_dynamic_ci.iloc[:, 1], color='k', alpha=.25)
ax.fill_betweenx(ax.get_ylim(), pd.to_datetime('2017-01-01'), y.index[-1],
               alpha=.1, zorder=-1)
ax.set_xlabel('Tahun')
ax.set_ylabel('Penumpang (Ribu)')
plt.legend()
plt.show()
```



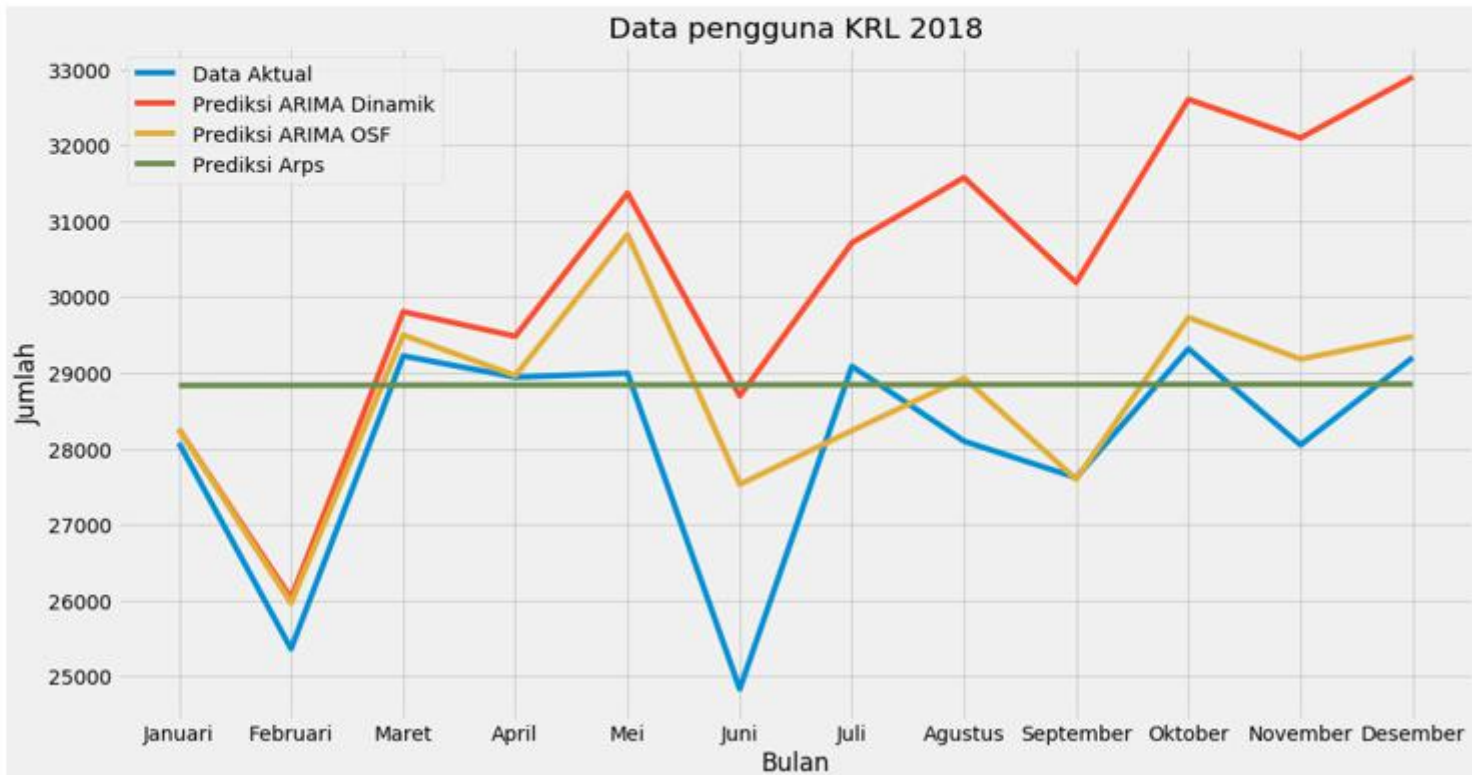
code untuk
plotting hasil
prediksi metode
Dynamic Forecast

grafik
perbandingan data
prediksi ARIMA
Dynamic Forecast
dengan data
aktual (2018)

Re-Implementation

Method and Code

```
In [115]: plt.figure(figsize=(15,8))
plt.plot(nama_bulan, y['2018-01-01:'], label='Data Aktual')
plt.plot(nama_bulan, pred_dynamic.predicted_mean, label='Prediksi ARIMA Dinamik')
plt.plot(nama_bulan, pred.predicted_mean, label='Prediksi ARIMA OSF')
plt.plot(nama_bulan, arps, label='Prediksi Arps')
plt.xlabel('Bulan')
plt.ylabel('Jumlah')
plt.title('Data pengguna KRL 2018')
plt.grid(True)
plt.legend()
plt.show()
```



code untuk
plotting
grafik
keseluruhan

grafik perbandingan
data prediksi ARIMA
dinamik, ARIMA OSF,
Arps Exponent, data
aktual (2018)



Thank you!

