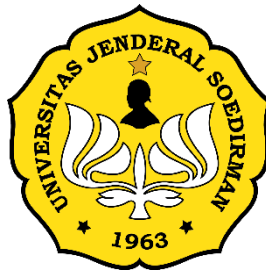


LAPORAN ANALISIS SISTEM

**ANALISIS *FORECASTING* YANG MEMPENGARUHI PRODUKSI PADI
DI PULAU SUMATERA INDONESIA MENGGUNAKAN METODE
LINEAR REGRESI DAN *AUTOREGRESSIVE INTEGRATED MOVING
AVERAGE* (ARIMA)**

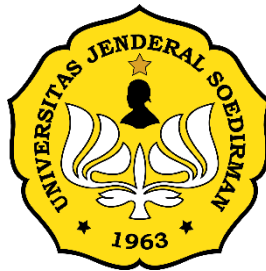


**Oleh:
Dhelya Apriliani Putri
NIM A1C021017**

**UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS PERTANIAN
PROGRAM STUDI TEKNIK PERTANIAN
PURWOKERTO
2023**

LAPORAN ANALISIS SISTEM

ANALISIS *FORECASTING* YANG MEMPENGARUHI PRODUKSI PADI DI PULAU SUMATERA INDONESIA MENGGUNAKAN METODE LINEAR REGRESI DAN *AUTOREGRESSIVE INTEGRATED MOVING AVERAGE* (ARIMA)



Oleh:
Dhelya Apriliani Putr
NIM A1C021017

**Diajukan Sebagai Salah Satu Penyetaraan Kegiatan Pada Mata Kuliah
Analisis Sistem Pada Pendidikan Strata Satu Program Studi Teknik
Pertanian Fakultas Pertanian Universitas Jenderal Soedirman**

**UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS PERTANIAN
PROGRAM STUDI TEKNIK PERTANIAN
PURWOKERTO
2023**

DAFTAR ISI

	Halaman
DAFTAR ISI	i
DAFTAR GAMBAR	ii
I. PENDAHULUAN	1
A. LATAR BELAKANG	1
B. TUJUAN	2
II. TINJAUAN PUSTAKA	3
III. METODE	4
IV. HASIL	6
A. Persiapan Data	6
B. Import Pustaka	6
C. Data Loading	7
D. Data Cleaning	8
E. Exploratory Data Analysis (EDA)	9
H. Data <i>Preprocessing</i>	19
I. Analisis <i>Forecasting</i>	20
V. PENUTUP	27
A. Kesimpulan	27
DAFTAR PUSTAKA	29
LAMPIRAN	30

DAFTAR GAMBAR

Gambar	Halaman
1. Data produksi padi.	6
2. Pustaka yang akan digunakan.	6
3. Menampilkan data frame teratas dan terbawah.	7
4. Menampilkan jumlah kolom dan jumlah baris.	7
5. Menampilkan statistika deskriptif.	8
6. Check missing value, duplicated nilai, nilai unik, nilai kosong,	8
7. Menampilkan category unik pada kolom 'Provinsi' dan 'Tahun'.	9
8. Histogram distribusi produksi.	9
9. Transformasi jhonson, normal, dan log normal pada data frame.	10
10. Distribusi independen.	10
11. Nilai skew pada kelembapan, suhu rata-rata, luas panen, dan	11
12. Transformasi jhonson, normal, dan log normal pada kolom kelembapan.	11
13. Correlation matrix.	12
14. Rata-Rata Produksi Padi Setiap Provinsi di Pulau Sumatera.	13
15. Pertumbuhan Produksi Tiap Tahun Berdasarkan Provinsi.	13
16. Coding visualisasi histogram terhadap luas panen, curah hujan,	14
17. Histogram luas panen.	14
18. Histogram curah hujan.	15
19. Histogram kelembapan.	15
20. Histogram suhu rata-rata.	16
21. Coding untuk menampilkan hasil hubungan antara variabel dengan	16
22. Plot antara luas panen dan produksi padi.	17
23. Plot antara curah hujan dan produksi padi.	17
24. Plot antara kelembapan dan produksi padi.	18
25. Plot antara suhu rata-rata dan produksi padi.	18
26. Bar plot rata rata pendapatan petani padi di pulau Sumatera.	19
27. Output data preprocessing.	20

28. Rumus min-max scaling.	20
29. Output model arima forecast.....	23
30. Ouput model linear regresi dan arima forest.....	25

I. PENDAHULUAN

A. LATAR BELAKANG

Indonesia dijuluki sebagai negara agraris karena mayoritas penduduknya bergantung pada sektor pertanian, hal ini didasarkan oleh Badan Pusat Statistik (2019) yang mengatakan sebanyak 33,4 juta penduduk Indonesia bermata pencaharian petani. Selain itu, luas lahan pertanian di Indonesia mencapai 10,65 HA, 7,46 Ha untuk sawah dan 35,19 untuk kebun ladang, tegal, dsb.

Berbagai macam jenis komoditas pertanian yang tersebar di Indonesia, padi merupakan salah satu komoditas yang paling banyak diusahakan oleh masyarakat, hal tersebut karena padi akan diolah menjadi beras yang merupakan makanan pokok masyarakat Indonesia. Meskipun begitu, meningkatnya konsumsi beras dari tahun ke tahun tidak berbanding lurus dengan pendapatan perekonomian di Indonesia. Peningkatan jumlah penduduk yang kian meningkat berpotensi mengganggu ketahanan pangan di Indonesia, maka dari itu untuk menambah pemasok beras dan menstabilkan harga beras, Indonesia menerapkan kebijakan impor. Menurut Badan Pusat Statistik (BPS), 2008 menjadi awal Indonesia melakukan impor beras. Namun, pada tahun 2016 hingga 2017 pemerintah melakukan pemberhentian sementara untuk impor beras karena stok beras dalam negeri telah terpenuhi atau tercukupi (swasembada beras).

Pulau Sumatera merupakan salah satu provinsi dengan penduduk terbanyak bermata pencaharian sebagai petani yang memiliki lebih dari 50% lahan pertanian di setiap provinsinya. Tingginya produksi beras di Sumatera Utara menyebabkan Sumatera Utara menduduki posisi ke-7 sebagai provinsi penghasil beras terbanyak di Indonesia. Namun, hasil pertanian di Sumatera sangat rentan terhadap perubahan iklim karena berpotensi pada pola dan waktu tanam, produksi dan kualitas hasil tanam. Peningkatan suhu bumi hingga ke pemanasan global juga akan mempengaruhi pola presipitasi, turunnya air hujan, kelembaban tanah, dan fluktuasi iklim yang secara keseluruhan akan mengancam keberhasilan dalam

produksi komoditas pertanian. Maka dari itu untuk mendorong tujuan pembangunan berkelanjutan atau SDGS, sebagai mahasiswa data scientist yang bergerak dibidang pertanian perlu membangun model prediktif dari masalah tersebut. Metode regresi yang digunakan adalah metode supervised learning menggunakan python anaconda.

Dataset yang digunakan yaitu dari tahun 1993 sampai 2020. Data meliputi provinsi, tahun, produksi, luas panen, curah hujan, kelembapan, dan suhu rata-rata. Dataset hasil produksi tahunan dan luas panen bersumber dari website BPS dan data perubahan kelembaban dan suhu rata rata bersumber dari website BMKG. Data tersebut kemudian di rangkum oleh website kaggle.

B. TUJUAN

1. Mengetahui hasil visualisasi terhadap variabel data *frame*
2. Mengetahui cara penerapan metode Regresi Linear dan ARIMA
3. Mengetahui bagaimana hubungan antara metode Regresi Linear dan ARIMA
4. Mengetahui hasil evaluasi penerapan metode Regresi Linear dan ARIMA

II. TINJAUAN PUSTAKA

Bahasa pemrograman Python, yang sangat populer saat ini, pertama kali ditemukan oleh Guido van Rossum di Stichting Mathematisch Centrum (CWI), Amsterdam pada tahun 1991 (Awangga *et al.*, 2019). Python adalah bahasa pemrograman yang menggunakan interpreter untuk menjalankan kode programnya. Interpreter tersebut dapat menerjemahkan kode secara langsung, dan Python dapat dijalankan di berbagai platform seperti Windows, Linux, dan lain-lain. Python mengadopsi paradigma pemrograman dari beberapa bahasa lain, termasuk paradigma pemrograman prosedural seperti bahasa C, pemrograman berorientasi objek seperti Java, dan bahasa fungsional seperti Lisp. Kombinasi paradigma ini memudahkan para programmer dalam mengembangkan berbagai proyek menggunakan Python (Rahman *et al.*, 2023).

Python sendiri sering digunakan sebagai saran untuk analisis data, metode analisis yang sering digunakan salah satunya adalah *forecasting*. Metode Forecast adalah prediksi dari beberapa peristiwa yang kemungkinan dapat terjadi di masa depan. Prediksi merupakan masalah penting yang mencakup berbagai bidang termasuk bisnis dan industri, pemerintahan, ekonomi, ilmu lingkungan, kedokteran, ilmu sosial, politik, dan keuangan. Proses prediksi diklasifikasikan menjadi prediksi jangka pendek, prediksi jangka menengah, dan prediksi jangka panjang. *Autoregressive Integrated Moving Average (ARIMA) models* atau *Box-Jenkins models* merupakan gabungan dari *autoregressive models* dengan *moving average models* (Hyndman *et al.*, 2018)

ARIMA merupakan kelas model yang menjelaskan deret waktu tertentu berdasarkan nilai dari masa lalu yang dimiliki, yaitu lags dirinya sendiri dan lags dari prediksi yang error. Model ARIMA memiliki 3 *hyperparameter* yaitu: p, d, dan q. yaitu istilah yang merujuk pada “Autoregressive” (AR), q yaitu istilah yang merujuk pada “Moving Average” (MA), dan d yaitu istilah yang merujuk pada “Differencing” (Tanuwidjaja, 2022).

III. METODE

A. Pengumpulan Data

Langkah awal pada penelitian ini yaitu melakukan studi literatur dengan melakukan pengumpulan data faktor yang mempengaruhi produksi padi di Pulau Sumatera yang diperoleh dari website kaggle. Data tersebut mencakup 6 variabel yaitu provinsi, tahun, produksi, luas panen, curah hujan, kelembaban, suhu rata-rata.

B. *Data Cleaning dan Exploration Data*

Pengolahan data diawali dengan mengumpulkan pustaka yang digunakan setelah itu membaca data set dan kemudian dilakukan pembersihan data (*data cleaning*) untuk memperbaiki atau menangani data yang rusak dan kemudian melakukan uji standarisasi dan outliers untuk menangani data yang sudah dibersihkan (*data cleaning*). Tahap selanjutnya yaitu melakukan visualisasi data terhadap data *frame* yang digunakan

C. Implementasi *Forecasting*

Pada data *frame* tanaman padi di Sumatera memiliki 8 *attribute* yang meliputi provinsi, tahun, produksi, luas panen, curah hujan, kelembapan, dan suhu rata-rata. *Attribute* tersebut bisa menjadi fokus studi kasus dalam melakukan *forecasting* terkait produksi padi di masa depan. Mengingat fluktuasi ekstrem curah hujan atau rendahnya tingkat curah hujan yang tercatat pada data, peramalan produksi padi menjadi krusial untuk pengelolaan pertanian yang berkelanjutan.

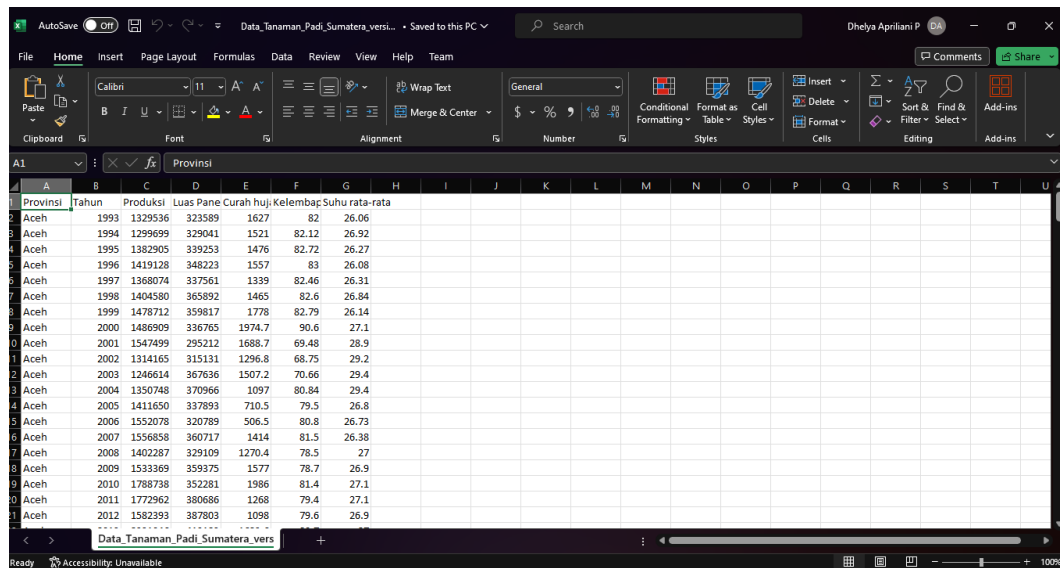
Dalam implementasi forecasting dibantu dengan 2 metode pemodelan yaitu regresi linear dan *Autoregressive Integrated Moving Average* (ARIMA). Sebelum

dimasukan kedalam model, *attribute* akan di normalisasikan terlebih dahulu menggunakan Min-Max Scalling

IV. HASIL

A. Persiapan Data

Hasil proses studi literatur dan pengolahan data yang digunakan disimpan dalam bentuk file Microsoft Excel dengan format CSV. Data yang diolah sebanyak 224 baris dan 7 kolom.



Provinsi	Tahun	Produksi	Luas Pane	Curah hujan	Kelembapan	Suhu rata-rata
Aceh	1993	1329536	323589	1627	82	26.06
Aceh	1994	1299699	329041	1521	82.12	26.92
Aceh	1995	1382905	339253	1476	82.72	26.27
Aceh	1996	1419128	348223	1557	83	26.08
Aceh	1997	1368074	337561	1339	82.46	26.31
Aceh	1998	1404580	365892	1465	82.6	26.84
Aceh	1999	1478712	359817	1778	82.79	26.14
Aceh	2000	1486909	336765	1974.7	90.6	27.1
Aceh	2001	1547499	295212	1688.7	69.48	28.9
Aceh	2002	1314165	315131	1296.8	68.75	29.2
Aceh	2003	1246614	367636	1507.2	70.66	29.4
Aceh	2004	1350748	370966	1097	80.84	29.4
Aceh	2005	1411650	337893	710.5	79.5	26.8
Aceh	2006	1552078	320789	506.5	80.8	26.73
Aceh	2007	1556858	360717	1414	81.5	26.38
Aceh	2008	1402287	329109	1270.4	78.5	27
Aceh	2009	1533369	359375	1577	78.7	26.9
Aceh	2010	1788738	352281	1986	81.4	27.1
Aceh	2011	1772962	380686	1268	79.4	27.1
Aceh	2012	1582393	387803	1098	79.6	26.9

Gambar 1. Data produksi padi.

B. Import Pustaka

Proses awal dalam melakukan analisis forecasting diperlukan untuk menentukan pustaka yang digunakan, dan pustaka yang dibutuhkan yaitu

Import Pustaka

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import plotly.express as px
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', None)

from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from keras.models import Sequential, load_model
from tensorflow.keras.optimizers import Adam
from keras.layers import Dense, LSTM, Dropout, BatchNormalization
from tensorflow.keras.metrics import RootMeanSquaredError
from tensorflow.keras.callbacks import TensorBoard
```

Gambar 2. Pustaka yang akan digunakan.

C. Data Loading

Data loading menjadi proses awal untuk melakukan eksplorasi, analisis dan membuat model machine learning. Langkah awal pada data loading yaitu import data frame :

```
df = pd.read_csv('Data_Tanaman_Padi_Sumatera_version_1.csv')
```

Kemudian menampilkan 5 baris pertama dan 5 baris terbawah dari data frame serta menampilkan jenis tipe data dan jumlah baris kolom. Setelah itu untuk mengetahui stastika dari data frame perlu `df.describe()`. Hal tersebut berfungsi untuk mendapatkan gambaran terkait struktur dari data frame yang digunakan dan mengambil kesimpulan mengenai rata-rata, standar deviasi, nilai minimum, kuartil, dan nilai maksimum. Gambar... merupakan output dari proses data loading.

```
In [3]: df.head()
```

Out[3]:

	Provinsi	Tahun	Produksi	Luas Panen	Curah hujan	Kelembapan	Suhu rata-rata
0	Aceh	1993	1329536.0	323589.0	1627.0	82.00	26.06
1	Aceh	1994	1299699.0	329041.0	1521.0	82.12	26.92
2	Aceh	1995	1382905.0	339253.0	1476.0	82.72	26.27
3	Aceh	1996	1419128.0	348223.0	1557.0	83.00	26.08
4	Aceh	1997	1368074.0	337561.0	1339.0	82.46	26.31

```
In [4]: df.tail()
```

Out[4]:

	Provinsi	Tahun	Produksi	Luas Panen	Curah hujan	Kelembapan	Suhu rata-rata
219	Lampung	2016	3831923.00	390799.00	2317.6	79.40	26.45
220	Lampung	2017	4090654.00	396559.00	1825.1	77.04	26.36
221	Lampung	2018	2488641.91	511940.93	1385.8	76.05	25.50
222	Lampung	2019	2164089.33	464103.42	1706.4	76.03	27.23
223	Lampung	2020	2604913.29	545149.05	2211.3	75.80	24.58

Gambar 3. Menampilkan data *frame* teratas dan terbawah.

```
In [5]: df.shape
```

Out[5]: (224, 7)

Gambar 4. Menampilkan jumlah kolom dan jumlah baris.

```
In [7]: df.describe()
```

```
Out[7]:
```

	Tahun	Produktel	Luas Panen	Curah hujan	Kelembapan	Suhu rata-rata
count	224.000000	2.240000e+02	224.000000	224.000000	224.000000	224.000000
mean	2008.500000	1.679701e+08	374349.988920	2452.490759	80.948705	26.801964
std	8.095838	1.161387e+08	232751.161987	1031.972825	4.878680	1.197041
min	1993.000000	4.293800e+04	63142.040000	222.500000	54.200000	22.190000
25%	1999.750000	5.488570e+05	146919.500000	1703.525000	78.975000	26.177500
50%	2008.500000	1.687773e+08	373551.500000	2315.700000	82.375000	26.730000
75%	2013.250000	2.436851e+08	514570.250000	3039.700000	84.000000	27.200000
max	2020.000000	4.881089e+08	872737.000000	5522.000000	90.600000	29.850000

Gambar 5. Menampilkan statistika deskriptif.

Berdasarkan analisis data loading diatas menunjukkan bahwa data faktor pengaruh produksi padi di Pulau Sumatera mempunyai 224 baris 7 kolom dengan tipe data 4 tipe data float64, 1 tipe data int64, 1 tipe data object. Distribusi statistika diatas menunjukkan bahwa hasil panen dari 8 provinsi mencapai rata rata 1679700.887 ton dengan rata-rata luas lahan pertanian adalah 374.350 hektar. Mean dan median di atas menunjukkan bahwa setiap feature relatif sama yang artinya berdistribusi normal.

D. Data Cleaning

Pada tahap *data cleaning* dilakukan indentifikasi terhadap isi dataframe yang digunakan. Proses identifikasi diantaranya yaitu nilai yang hilang, nilai yang sama, dan nilai yang unik pada kolom category atau object. Gambar merupakan *ouput* dari proses *data cleaning*.

```
In [12]: cek = pd.DataFrame({
          'Data Kosong': df.isnull().sum(),
          'Data Duplikat': df.duplicated().sum(),
          'Data NaNN': df.isna().sum(),
          'Data Unique': df.nunique(),
          'Type Data': df.dtypes
        })
cek
```

```
Out[12]:
```

	Data Kosong	Data Duplikat	Data NaNN	Data Unique	Type Data
Provinsi	0	0	0	8	object
Tahun	0	0	0	28	int64
Produksi	0	0	0	224	float64
Luas Panen	0	0	0	224	float64
Curah hujan	0	0	0	220	float64
Kelembapan	0	0	0	180	float64
Suhu rata-rata	0	0	0	136	float64

Gambar 6. Check missing value, duplicated nilai, nilai unik, nilai kosong, dan tipe data.

Output dari *data cleaning* diatas menunjukkan bahwa pada data *frame* yang digunakan tidak mengandung nilai yang kosong, tidak memiliki kesamaan nilai (*duplicated values*), tidak mengandung nilai hilang atau data NaNN (*missing values*), dan mengandung beberapa data unik salah satunya adalah variabel Provinsi dengan 8 data unik dan variable Tahun dengan 28 data unik. Hal tersebut mengindikasi bahwa data *frame* ini mencakup jumlah produksi dari 8 provinsi dengan rentang waktu selama 28 tahun (1993 – 2020).

```
In [13]: df['Provinsi'].unique()

Out[13]: array(['Aceh', 'Sumatera Utara', 'Sumatera Barat', 'Riau', 'Jambi',
               'Sumatera Selatan', 'Bengkulu', 'Lampung'], dtype=object)

In [14]: df['Tahun'].unique()

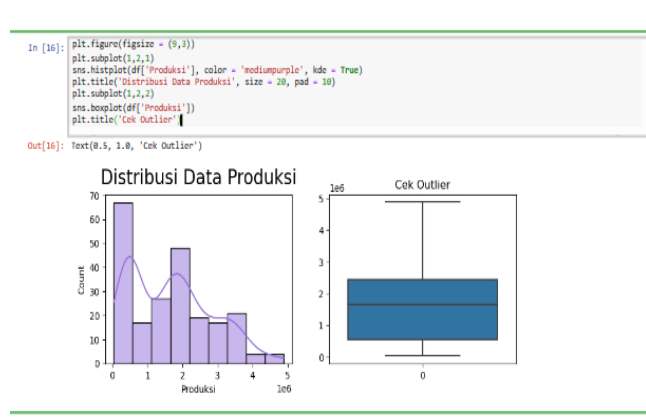
Out[14]: array([1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003,
               2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014,
               2015, 2016, 2017, 2018, 2019, 2020], dtype=int64)
```

Gambar 7. Menampilkan *category* unik pada kolom 'Provinsi' dan 'Tahun'.

E. Exploratory Data Analysis (EDA)

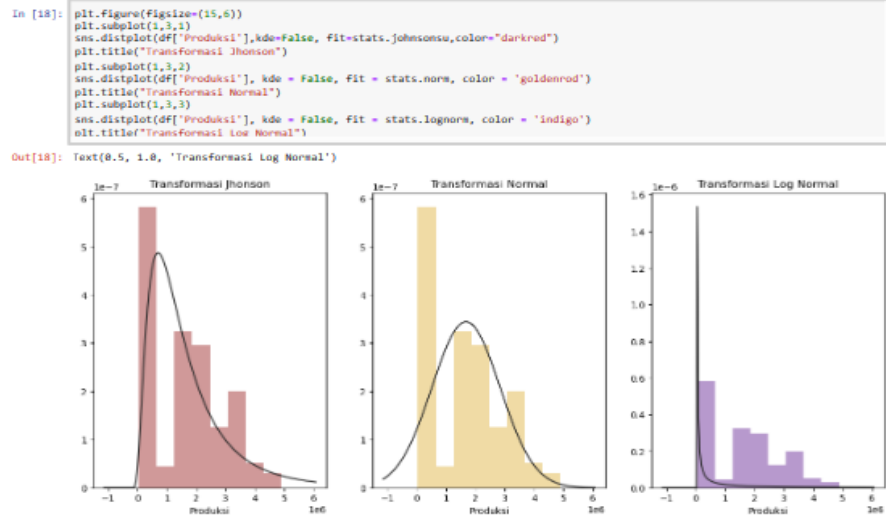
Exploratory Data Analysis (EDA) ini merupakan pendekatan analisis data guna memahami data lebih mendalam. Pada tahap ini dilakukannya visualisasi data yang meliputi

1. Cek Distribusi
 - a. Cek distribusi pada data target



Gambar 8. Histogram distribusi produksi.

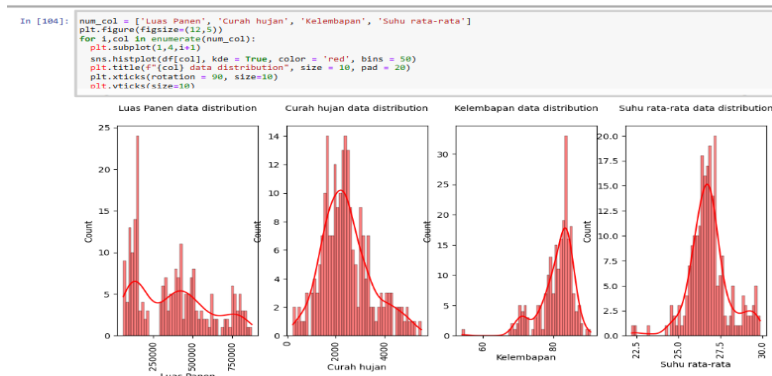
Berdasarkan hasil *outlier* data produksi diatas menunjukan bahwa tidak ada *outlier* dari variabel *output* yang menandakan bahwa data target berdistribusi normal. Setelah itu, cek tranformasi untuk mengetahui jika terdapat transformasi yang dapat lebih menormalkan data target.



Gambar 9. Transformasi jhonson, normal, dan log normal pada data frame produksi.

Plot diatas memvisualisasikan bahwa tranformasi menggunakan metode normal menunjukan distribusi yang cukup normal.

b. Cek distribusi pada data independen



Gambar 10. Distribusi independen.

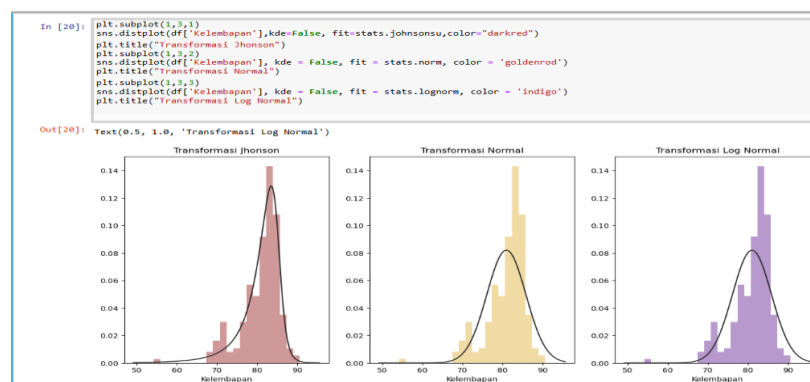
```
In [103]: df[num_col].skew().sort_values().to_frame().rename(columns = {'0':'Skew'})
```

Out[103]:

	Skew
Kelembapan	-1.487425
Suhu rata-rata	0.061508
Luas Panen	0.428898
Produksi	0.549053
Curah hujan	0.631927

Gambar 11. Nilai *skew* pada kelembapan, suhu rata-rata, luas panen, dan produksi.

Plot diatas memvisualisasikan bahwa variabel diatas sudah berdistribusi normal namun pada variabel kelembapan menunjukan jika skewness negatif yang mana harus dilakukan transformasi lebih lanjut.

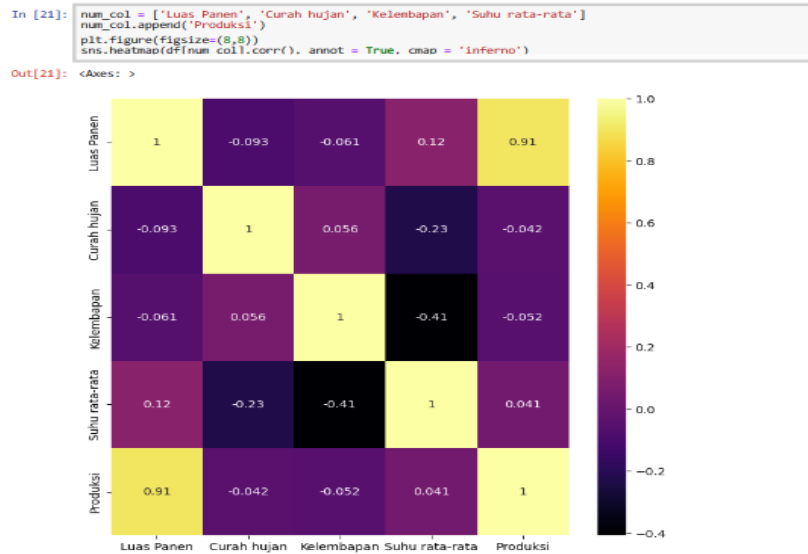


Gambar 12. Transformasi jhonson, normal, dan log normal pada kolom kelembapan.

F. Correlation Matrix

Correlation matrix merupakan suatu matriks yang menunjukkan tingkat korelasi antar variabel pada dataset. Rentang nilai pada matriks yaitu -1 dan 1;

1. Nilai 1 yaitu korelasi positif sempurna (korelasi positif penuh)
2. Nilai -1 yaitu korelasi negatif sempurna (korelasi negatif penuh)
3. Nilai 0 yaitu tidak adanya korelasi (korelasi 0)

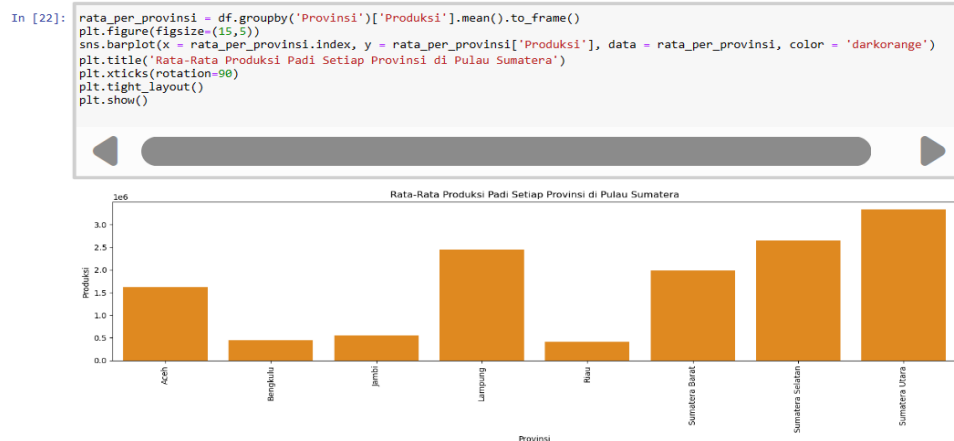


Gambar 13. *Correlation matrix.*

Correlation matrix diatas mengandung nilai 1 yang artinya memiliki korelasi positif sempurna dan menunjukkan bahwa luas lahan pertanian dan suhu rata-rata memiliki korelasi positif, yang berarti bahwa ketika nilai luas lahan pertanian meningkat, produksi juga cenderung meningkat, meskipun pengaruh suhu tidak begitu signifikan. Di sisi lain, curah hujan dan kelembapan memiliki korelasi negatif, yang berarti bahwa ketika nilai kedua variabel ini menurun, produksi cenderung meningkat. Meskipun demikian, hubungan ini tidak begitu kuat, dan perubahan dalam grafik tidak terlalu signifikan.

G. Visualisasi Data

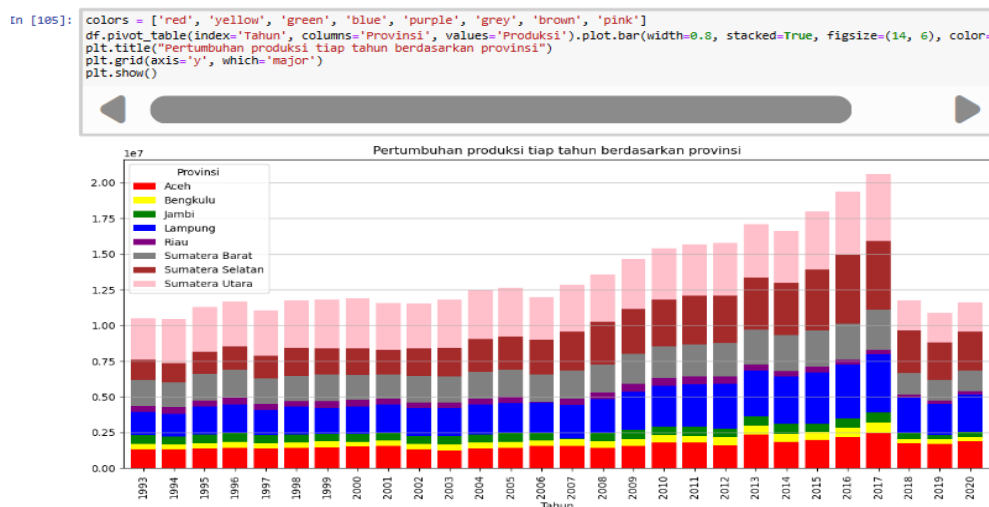
1. Rata-Rata Produksi Padi Setiap Provinsi di Pulau Sumatera



Gambar 14. Rata-Rata Produksi Padi Setiap Provinsi di Pulau Sumatera.

Visualisasi diatas menunjukan bahwa produksi padi tertinggi berada di provinsi Sumatera Utara yaitu sebanyak 3 ton sementara produksi padi terendah berada di provinsi Bengkulu dan Riau yang tidak mencapai 0,5 ton.

2. Pertumbuhan Produksi Tiap Tahun Berdasarkan Provinsi



Gambar 15. Pertumbuhan Produksi Tiap Tahun Berdasarkan Provinsi.

Visualisasi diatas menunjukan bahwa peningkatan produksi padi dimulai sejak tahun 2003 yaitu sebanyak 11823024 ton hingga pada tahun 2017 yang merupakan tahun dengan produksi padi tertinggi yang mencapai berat lebih dari 20586773,50 ton namun sempat mengalami penurunan pada tahun 2006 sekitar 12001280,00 ton dan kembali mengalami penurunan drastis sebanyak 1 ton pada tahun 2018 yaitu hanya 10881099,49 ton.

Penurunan produksi padi tersebut dapat dipengaruhi oleh perubahan iklim yang meliputi tingkat curah hujan, tingkat kelembapan, dan tingkat perubahan suhu di Pulau Sumatera.

```
In [137]:
luas_panen = df.groupby('Tahun')['Luas Panen'].mean().to_frame()
plt.figure(figsize=(15,5))
sns.barplot(x = luas_panen.index, y = luas_panen['Luas Panen'], data = luas_panen, color = 'lightblue')
plt.title('Luas Panen')

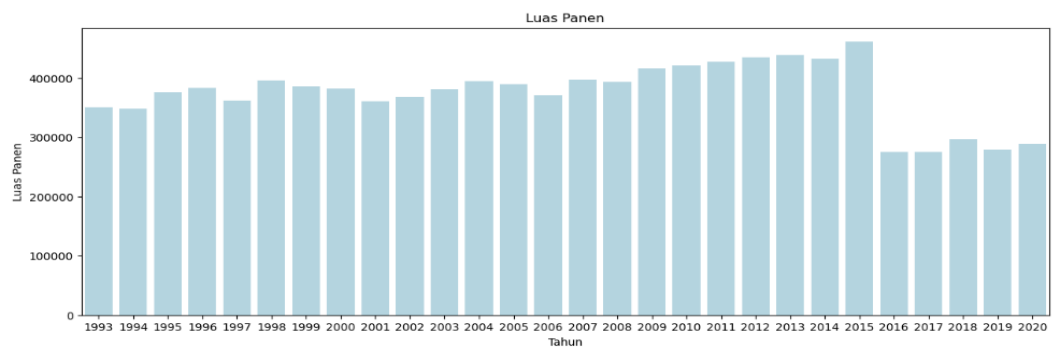
curah_hujan = df.groupby('Tahun')['Curah hujan'].mean().to_frame()
plt.figure(figsize=(15,5))
sns.barplot(x = curah_hujan.index, y = curah_hujan['Curah hujan'], data = curah_hujan, color = 'lightblue')
plt.title('Curah hujan')

Kelembaban = df.groupby('Tahun')['Kelembapan'].mean().to_frame()
plt.figure(figsize=(15,5))
sns.barplot(x = Kelembaban.index, y = Kelembaban['Kelembapan'], data = Kelembaban, color = 'lightblue')
plt.title('Kelembapan')

suhu_rata = df.groupby('Tahun')['Suhu rata-rata'].mean().to_frame()
plt.figure(figsize=(15,5))
sns.barplot(x = suhu_rata.index, y = suhu_rata['Suhu rata-rata'], data = suhu_rata, color = 'lightblue')
plt.title('Suhu rata-rata')

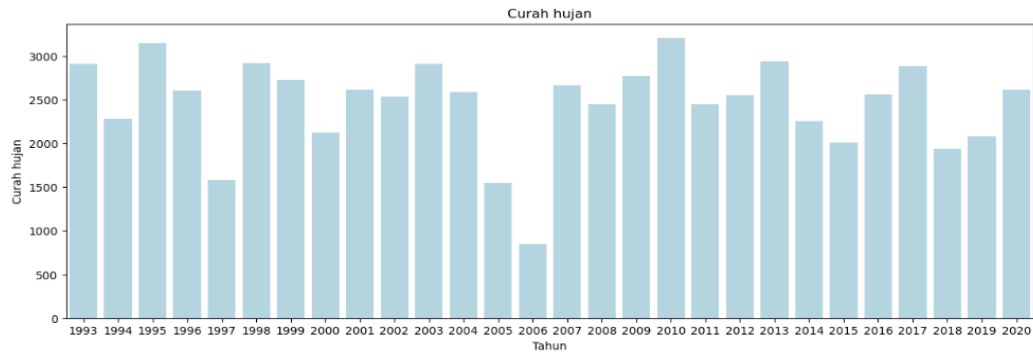
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Gambar 16. Coding visualisasi histogram terhadap luas panen, curah hujan, kelembapan, dan suhu rata rata.



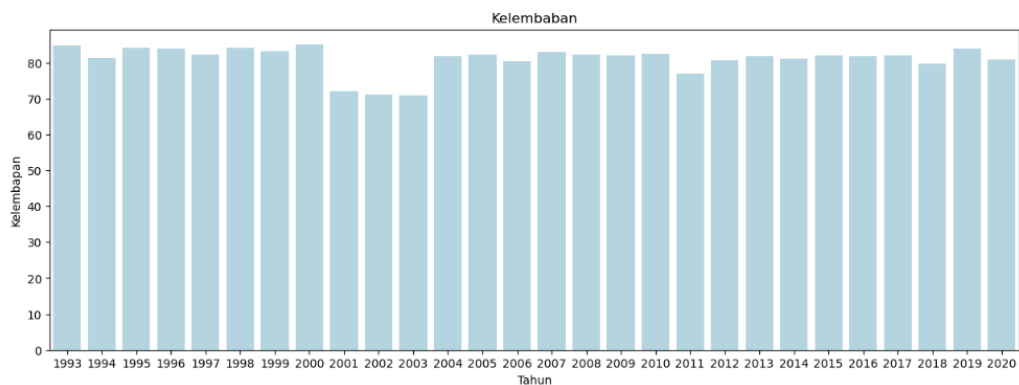
Gambar 17. Histogram luas panen

Berdasarkan histogram luas panen diatas menunjukan bahwa luas area panen terbesar yaitu pada tahun 2015 dengan rata rata sebesar 461121 Ha dan luas area panen terkecil yaitu pada tahun 2016 sebesar 274593 Ha. Ketika mengalami tingginya produksi padi pada tahun 2017 rata rata luas area panennya yaitu 275097 Ha sementara ketika terjadi penurunan produksi padi secara drastis pada tahun 2018 yaitu sebesar 295975 Ha.



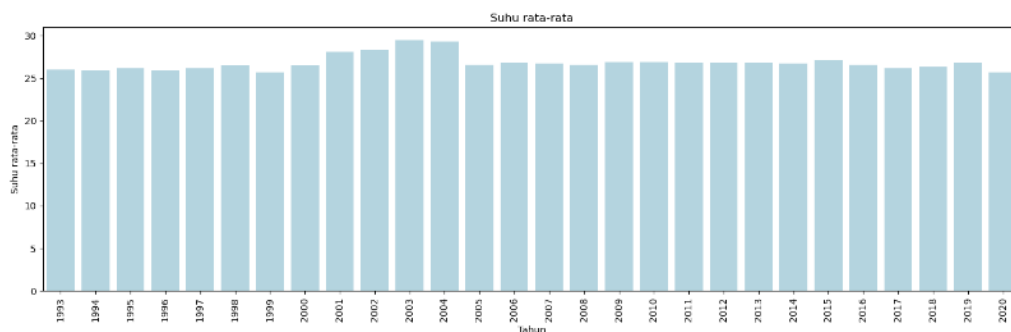
Gambar 18. Histogram curah hujan.

Berdasarkan histogram curah hujan diatas menunjukkan bahwa curah hujan tertinggi yaitu pada tahun 2010 dengan rata rata curah hujan sebesar 3205 mm dan tingkat curah hujan terendah yaitu pada tahun 2006 sebesar 847 mm. Ketika mengalami tingginya produksi padi pada tahun 2017 rata rata curah hujan yang terjadi yaitu 2886 mm sementara ketika terjadi penurunan produksi padi pada tahun 2018 yaitu sebesar 1936 mm.



Gambar 19. Histogram kelembaban.

Berdasarkan histogram kelembaban diatas menunjukkan bahwa kelembaban tertinggi yaitu pada tahun 2000 dengan rata rata kelembaban sebesar 84 % dan tingkat kelembaban terendah yaitu pada tahun 2003 sebesar 71 %. Ketika mengalami tingginya produksi padi pada tahun 2017 rata rata kelembaban yang terjadi yaitu 82 % sementara ketika terjadi penurunan produksi padi pada tahun 2018 yaitu sebesar 80 %.



Gambar 20. Histogram suhu rata-rata.

Berdasarkan histogram suhu rata rata diatas menunjukan bahwa rata rata suhu tertinggi yaitu pada tahun 2003 dengan rata rata suhu sebesar 29°C dan tingkat kelembaban terendah yaitu pada tahun 2003 sebesar 70%. Ketika mengalami tingginya produksi padi pada tahun 2017 rata rata kelembaban yang terjadi yaitu 81% sementara ketika terjadi penurunan produksi padi pada tahun 2018 yaitu sebesar 79%.

3. Hubungan antara Variabel dengan Produksi Padi

```
n [487]: fig, axes = plt.subplots(2, 2, figsize=(9, 9))

# Hubungan antara Luas panen dan produksi padi
sns.regplot(ax=axes[0, 0], x=df['Luas Panen'], y=df['Produksi'], scatter_kws={'alpha':0.5}, line_kws={'color':'green'})
axes[0, 0].set_title('Luas Pendapatan vs Produksi Padi')
axes[0, 0].set_xlabel('Luas Panen (hektar)')
axes[0, 0].set_ylabel('Produksi Padi (ton)')
axes[0, 0].tick_params(labelsize=8)

# Hubungan antara curah hujan dan produksi padi
sns.regplot(ax=axes[0, 1], x='Curah hujan', y='Produksi', data=df, scatter_kws={'alpha':0.5}, line_kws={'color':'red'})
axes[0, 1].set_title('Curah Hujan vs Produksi Padi')
axes[0, 1].set_xlabel('Curah Hujan (mm)')
axes[0, 1].set_ylabel('Produksi Padi (ton)')
axes[0, 1].tick_params(labelsize=8)

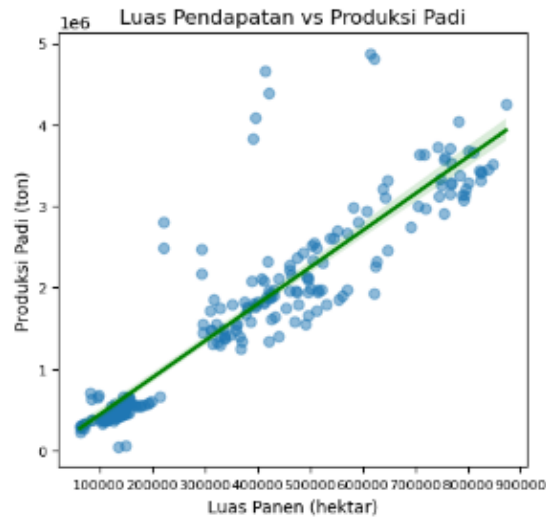
# Hubungan antara kelembapan dan produksi padi
sns.regplot(ax=axes[1, 0], x=df['Kelembapan'], y=df['Produksi'], scatter_kws={'alpha':0.5}, line_kws={'color':'blue'})
axes[1, 0].set_title('Kelembapan vs Produksi Padi')
axes[1, 0].set_xlabel('Kelembapan (%)')
axes[1, 0].set_ylabel('Produksi Padi (ton)')
axes[1, 0].tick_params(labelsize=8)

# Hubungan antara suhu rata-rata dan produksi padi
sns.regplot(ax=axes[1, 1], x=df['Suhu rata-rata'], y=df['Produksi'], scatter_kws={'alpha':0.5}, line_kws={'color':'purple'})
axes[1, 1].set_title('Suhu Rata-rata vs Produksi Padi')
axes[1, 1].set_xlabel('Suhu Rata-rata (°C)')
axes[1, 1].set_ylabel('Produksi Padi (ton)')
axes[1, 1].tick_params(labelsize=8)

plt.tight_layout()
plt.show()
```

Gambar 21. Coding untuk menampilkan hasil hubungan antara variabel dengan produksi padi.

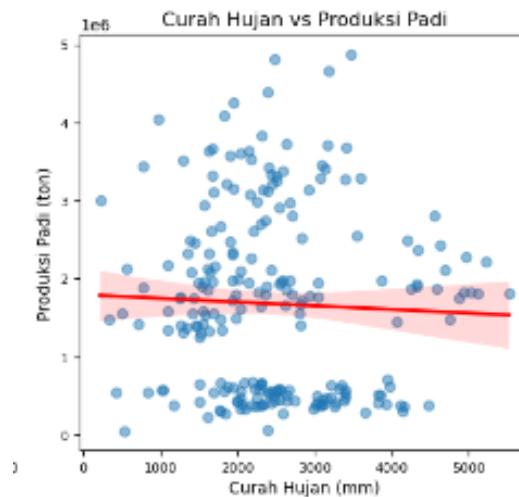
a. Luas panen dan produksi padi



Gambar 22. Plot antara luas panen dan produksi padi.

Korelasi positif antara luas panen dan produksi menunjukkan bahwa peningkatan luas tanam padi dapat meningkatkan produksi secara keseluruhan.

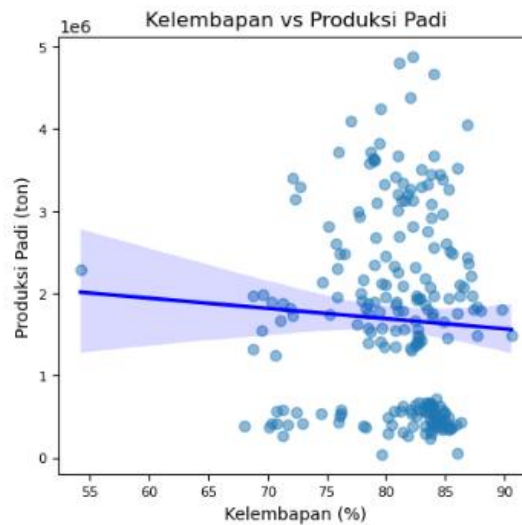
b. Curah hujan dan produksi padi



Gambar 23. Plot antara curah hujan dan produksi padi.

Penambahan plot hubungan curah hujan memperkuat korelasi positif antara curah hujan dan produksi beras, yang menunjukkan bahwa pasokan air yang cukup dari curah hujan bermanfaat bagi hasil panen padi.

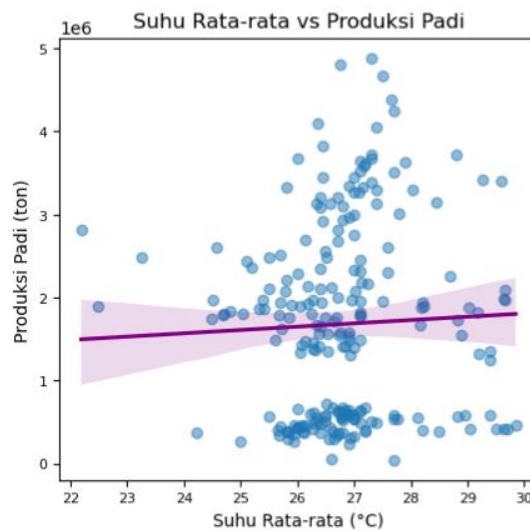
c. Kelembaban dan produksi padi



Gambar 24. Plot antara kelembaban dan produksi padi

Keberagaman dalam hubungan antara kelembaban dan produksi menunjukkan bahwa kelembaban saja bukan merupakan prediktor yang kuat terhadap tingkat produksi, kemungkinan besar disebabkan oleh kemampuan adaptasi beras terhadap tingkat kelembaban yang berbeda.

d. Suhu rata-rata dan produksi padi



Gambar 25. Plot antara suhu rata-rata dan produksi padi.

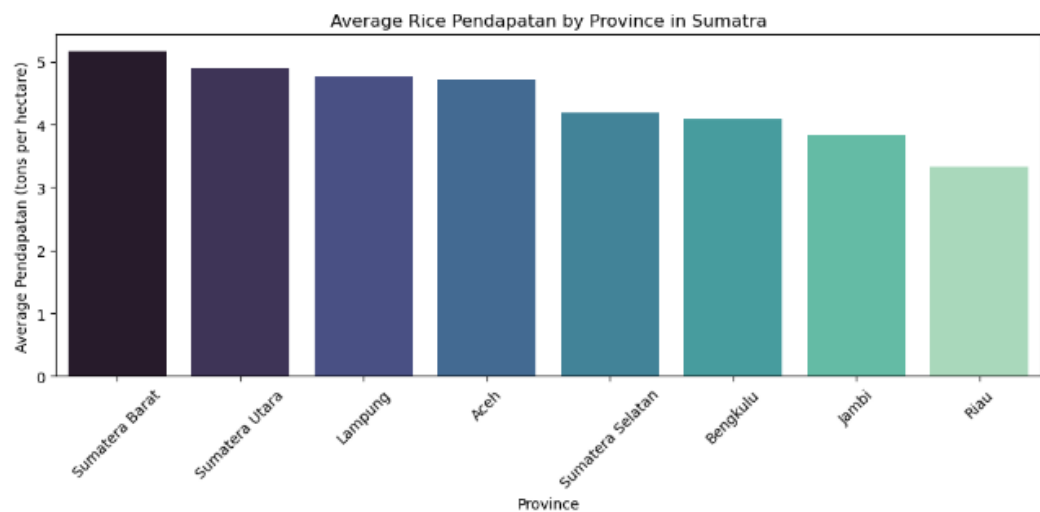
Hubungan antara suhu dan produksi nampaknya lemah, hal ini menunjukkan bahwa dalam kisaran suhu yang diamati, produksi beras tidak terlalu sensitif terhadap variasi suhu di Sumatera.

4. Pendapatan

Pendapatan dapat dihitung dengan membagi variable produksi dengan variabel luas panes :

$$df['Pendapatan'] = df['Produksi'] / df['Luas Panen']$$

Kemudian visualisasikan menggunakan barplot :



Gambar 26. Bar plot rata rata pendapatan petani padi di pulau Sumatera.

H. Data Preprocessing

Data preprocessing yang digunakan pada kasus ini yaitu simple imputer yang merupakan salah pustaka scikit learn yang dapat membantu untuk menggantikan missing values tertentu seperti mean, media, atau nilai konstan.

```
imputer = SimpleImputer(strategy='mean')
```

```
df[['Produksi', 'Luas Panen', 'Curah hujan', 'Kelembapan', 'Suhu rata-rata']] =  
imputer.fit_transform(df[['Produksi', 'Luas Panen', 'Curah hujan', 'Kelembapan',  
'Suhu rata-rata']])
```

Langkah awal pada preprocessing ini yaitu menormalisasi data dengan mengabaikan kolom provinsi dan tahun menggunakan *Min-Max Scaling*

```
numeric_data = df.drop(['Provinsi', 'Tahun'], axis=1)
```



```
scaler = MinMaxScaler()
```

```
scaled_data = scaler.fit_transform(numeric_data)
```

```
normalized_data = pd.DataFrame(scaled_data, columns=numeric_data.columns)
print(normalized_data.head())
```

Menghasilkan output seperti dibawah :

	Produksi	Luas Panen	Curah hujan	Kelembapan	Suhu rata-rata	Pendapatan
0	0.265928	0.321700	0.265025	0.763736	0.505222	0.307331
1	0.259761	0.328435	0.245023	0.767033	0.617493	0.294469
2	0.276958	0.341048	0.236532	0.783516	0.532637	0.304707
3	0.284445	0.352128	0.251816	0.791209	0.507833	0.304627
4	0.273893	0.338958	0.210680	0.776374	0.537859	0.302802

Gambar 27. Output data preprocessing.

I. Analisis Forecasting

1. Regresi Linear

Tahapan awal dalam melakukan uji model regresi linear yaitu menormalisasikan data menggunakan metode *Min-Max Scaling*

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Gambar 28. Rumus min-max scaling.

Normalisasi menggunakan *Min-Max Scaling* merupakan metode yang berfungsi untuk merubah nilai dalam suatu data *frame* ke dalam rentang 0 dan 1, metode ini dapat menjaga proporsi nilai relatif serta dapat meningkatkan kinerja beberapa algoritma *machine learning*. Gambar () merupakan rumus *Min-Max Scaling* secara manual, namun jika perhitungannya melalui jupyter notebook yaitu dapat memasukan code sebagai berikut

Tahapan dari pembuatan model regresi linear yaitu

- a. Mengabaikan kolom 'Provinsi' dan 'Tahun' untuk normalisasi

```
numeric_data = df.drop(['Provinsi', 'Tahun'], axis=1)
```

- b. Normalisasi menggunakan *Min-Max Scaling*

```
scaler = MinMaxScaler()
```

```
scaled_data = scaler.fit_transform(numeric_data)
```

- c. Membuat *DataFrame* baru dengan data yang sudah dinormalisasi

```
normalized_data = pd.DataFrame(scaled_data,  
                                columns=numeric_data.columns)
```

- d. Memilih fitur (variabel independen) dan target (variabel dependen)

```
X = normalized_data.drop('Produksi', axis=1)
```

```
y = normalized_data['Produksi']
```

- e. Memisahkan data menjadi data latih dan data uji

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
                                                    random_state=42)
```

- f. Membuat model regresi linear

```
model = LinearRegression()
```

- g. Melatih model dengan data latih

```
model.fit(X_train, y_train)
```

- h. Membuat prediksi dengan data uji

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

- i. Evaluasi performa model

```
mse = mean_squared_error(y_test, y_pred)
```

```
r2 = r2_score(y_test, y_pred)
```

- j. Memisahkan data menjadi data latih dan data uji

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
                                                    random_state=42)
```

- k. Membuat model regresi linear

```
model = LinearRegression()
```

- l. Melatih model dengan data latih

```
model.fit(X_train, y_train)
```

- m. Membuat prediksi dengan data uji

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

- n. Evaluasi performa model

```
mse = mean_squared_error(y_test, y_pred)
```

```
r2 = r2_score(y_test, y_pred)
```

- o. Menampilkan hasil evaluasi

```
print(f'Mean Squared Error: {mse}')
```

```
print(f'R-squared: {r2}')
```

- p. Menampilkan koefisien dan intercept

```
print('Koefisien:', model.coef_)
```

```
print('Intercept:', model.intercept_)
```

Tahapan dari pembuatan model regresi linear diatas menghasilkan Mean Squared Error (MSE): MSE untuk model regresi linear adalah 0.00276.

2. Arima Forecast

Tahap - tahap dalam melakukan analisis *arima forecast*

- a. Mengabaikan kolom 'Provinsi' dan 'Tahun' untuk *forecasting*

```
time_series_data = df.groupby('Tahun')['Produksi'].sum().reset_index()
```

- b. *Split* data menjadi *train* dan *test*

```
train_data = time_series_data[:-10] # Menggunakan 10 tahun terakhir  
sebagai test
```

```
test_data = time_series_data[-10:]
```

- c. Membangun model ARIMA

```
p, d, q = 1, 1, 1 # Sesuaikan dengan parameter terbaik berdasarkan  
analisis atau tuning model
```

```
arima_model = ARIMA(train_data['Produksi'], order=(p, d, q))
```

```
arima_result = arima_model.fit()
```

- d. Melakukan *forecasting* untuk data *test*

```
forecast_values = arima_result.get_forecast(steps=len(test_data))
```

```
forecast_mean = forecast_values.predicted_mean
```

e. Menampilkan hasil forecasting

```
plt.figure(figsize=(10, 6))
```

```
plt.plot(train_data["Tahun"], train_data["Produksi"], label='Train Data')
```

```
plt.plot(test_data["Tahun"], test_data["Produksi"], label='Actual Data')
```

```
plt.plot(test_data["Tahun"], forecast_mean, label='ARIMA Forecast')
```

```
plt.title('ARIMA Forecasting for Padi Production')
```

```
plt.xlabel('Tahun')
```

```
plt.ylabel('Produksi')
```

```
plt.legend()
```

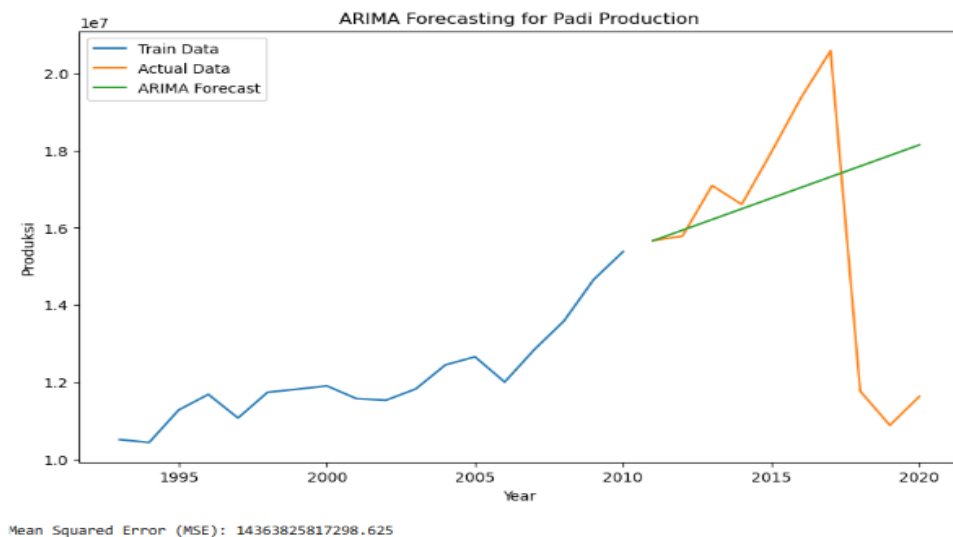
```
plt.show()
```

f. Mengukur performa model menggunakan MSE (Mean Squared Error)

```
mse = mean_squared_error(test_data["Produksi"], forecast_mean)
```

```
print(f'Mean Squared Error (MSE): {mse}')
```

Berdasarkan tahapan arima forest diatas menghasilkan grafik dan hasil untuk Mean Squared Error (MSE): MSE untuk model ARIMA adalah 14,360,487,067,739.84.



Gambar 29. Output model arima forecast.

3. Visualisasi hasil dari linear regresi dan arima *forest*

a. Model Regresi Linear

<code>numeric_data = df.drop(['Provinsi', 'Tahun'], axis=1)</code>
<code>scaler = MinMaxScaler()</code>
<code>scaled_data = scaler.fit_transform(numeric_data)</code>
<code>normalized_data = pd.DataFrame(scaled_data, columns = numeric_data.columns)</code>
<code>X = normalized_data.drop('Produksi', axis=1)</code>
<code>y = normalized_data['Produksi']</code>
<code>X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)</code>
<code>model_linear = LinearRegression()</code>
<code>model_linear.fit(X_train, y_train)</code>
<code>y_pred_linear = model_linear.predict(X_test)</code>
<code>mse_linear = mean_squared_error(y_test, y_pred_linear)</code>
<code>r2_linear = r2_score(y_test, y_pred_linear)</code>

b. Model ARIMA

<code>time_series_data = df.groupby('Tahun')['Produksi'].sum().reset_index()</code>
<code>train_data = time_series_data[:-10]</code>
<code>test_data = time_series_data[-10:]</code>
<code>p, d, q = 1, 1, 1</code>
<code>arima_model = ARIMA(train_data['Produksi'], order=(p, d, q))</code>
<code>arima_result = arima_model.fit()</code>

c. Melakukan forecasting untuk data test

<code>forecast_values = arima_result.get_forecast(steps=len(test_data))</code>
<code>forecast_mean_arima = forecast_values.predicted_mean</code>
<code>mse_arima = mean_squared_error(test_data['Produksi'], forecast_mean_arima)</code>

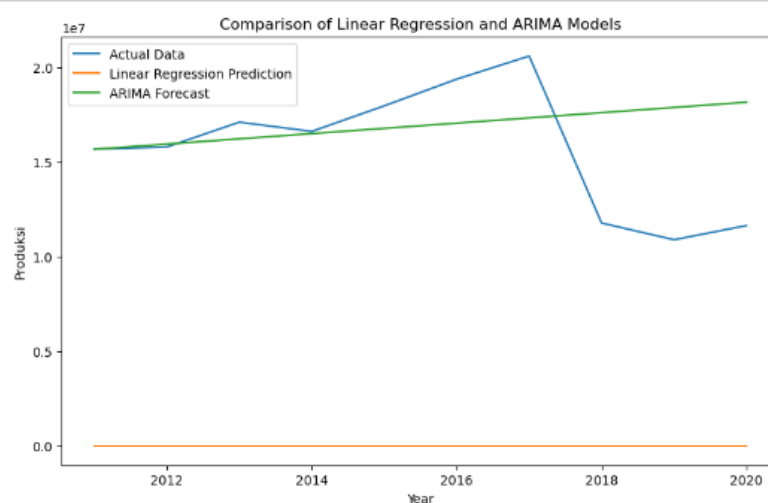
d. Visualisasi

```
plt.figure(figsize=(10, 6))
plt.plot(test_data['Tahun'], test_data['Produksi'], label='Actual Data')
plt.plot(test_data['Tahun'], y_pred_linear[:len(test_data)], label='Linear
Regression Prediction')
plt.plot(test_data['Tahun'], forecast_mean_arima, label='ARIMA
Forecast')
plt.title('Comparison of Linear Regression and ARIMA Models')
plt.xlabel('Year')
plt.ylabel('Produksi')
plt.legend()
plt.show()
```

e. Menampilkan hasil evaluasi

```
print('Linear Regression Model:')
print(f'Mean Squared Error: {mse_linear}')
print(f'R-squared: {r2_linear}')
print('\nARIMA Model:')
print(f'Mean Squared Error (MSE): {mse_arima}')
```

Hasil visualisasi dari 2 model diatas menunjukan grafik seperti gambar



Gambar 30. Ouput model linear regresi dan arima forest.

Linear Regression Model:

Mean Squared Error (MSE): MSE untuk model regresi linear adalah 0.0049. MSE mengukur seberapa dekat prediksi model dengan nilai aktual. Semakin rendah MSE, semakin baik modelnya. Dalam konteks ini, MSE yang rendah menunjukkan bahwa model regresi linear secara relatif baik dalam memprediksi produksi padi.

ARIMA Model:

Mean Squared Error (MSE): MSE untuk model ARIMA adalah 14,360,487,067,739.84. Nilai MSE yang sangat tinggi ini menunjukkan bahwa model ARIMA tidak cukup baik dalam memprediksi produksi padi berdasarkan data uji yang digunakan. MSE yang tinggi menandakan bahwa terdapat ketidakcocokan yang signifikan antara nilai aktual dan nilai prediksi.

V. PENUTUP

A. Kesimpulan

1. Hasil visualisasi data *frame*

Visualisasi pada data *frame* memberikan gambaran komprehensif tentang bagaimana berbagai faktor lingkungan berkorelasi dengan produksi padi di Sumatera. Curah hujan yang cukup dan lahan pertanian yang cukup tampaknya menjadi faktor kunci dalam memaksimalkan hasil panen padi, sementara kelembapan dan suhu dalam kisaran normal di wilayah tersebut mempunyai pengaruh yang kurang jelas. Wawasan ini dapat menjadi masukan bagi kebijakan pertanian yang berfokus pada pengelolaan air, penggunaan lahan, dan strategi adaptasi iklim.

2. Penerapan Regresi linear dan ARIMA :

Penerapan Regresi Linear pada kasus ini yaitu mampu untuk menangani variabel numerik seperti curah hujan, kelembapan, dan suhu rata-rata yang menjadi faktor pengaruh produksi padi, regresi linear juga memberikan interpretasi yang sederhana terhadap hubungan linier antara variabel-independen dan variabel dependen serta memberikan pemahaman tentang bagaimana setiap variabel berkontribusi terhadap produksi padi. Sementara penerapan ARIMA pada kasus ini yaitu mampu mendeteksi pola musiman atau tren jangka panjang yang muncul dari fluktuasi pengaruh yang ada pada data *frame*.

3. Hasil penerapan model Regresi Linear dan ARIMA

Hasil dari penerapan penggunaan model Regresi Linear dan ARIMA pada kasus ini mampu memberikan pendekatan secara holistik dan membantu dalam pemilihan model yang berpotensi untuk memanfaatkan kedua model tersebut.

4. Evaluasi

Hasil evaluasi analisis untuk kedua model (MSE dan *R-squared* untuk Regresi Linear, MSE untuk ARIMA) memberikan pemahaman tentang seberapa baik model-model tersebut dapat melakukan prediksi berdasarkan data yang ada. Dengan kombinasi kedua model tersebut dapat memberikan pendekatan yang cukup kuat untuk melakukan peramalan produksi padi di masa depan, dengan mempertimbangkan faktor-faktor eksternal dan pola waktu dalam data.

DAFTAR PUSTAKA

- Rahman, S., Sembiring, A., Siregar, D., Prahmana, I. G., Puspadini, R., & Zen, M. (2023). PYTHON: DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK. Penerbit Tahta Media.
- R. J. Hyndman dan G. Athanasopoulos. (2018). Forecasting: Principles and Practice, Australia: Monash University.
- Tanuwidjaja, K., & Widjaja, A. (2022). Prediksi dan Analisis Time Series pada Data COVID-19. Jurnal STRATEGI-Jurnal Maranatha, 4(1), 144-158.

LAMPIRAN

Hasil pengerjaan melalui jupyter notebook

[dhelyaapriliani/ANALISIS-SISTEM \(github.com\)](https://github.com/dhelyaapriliani/ANALISIS-SISTEM)