

**LAPORAN**  
**RENCANA TUGAS MAHASISWA (RTM) Ke-3**  
**MATA KULIAH ANALISIS DATA EKSPLORATIF**  
**“Statistika Deskriptif dan Grouping Dataset”**



**DISUSUN OLEH:**

Muhammad Aryasatya Nugroho ( 22083010085 )

**DOSEN PENGAMPU:**

Tresna Maulana Fahrudin S.ST., M.T. (NIP. 199305012022031007)

**PROGRAM STUDI SAINS DATA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR**  
**2023**

*Implementasikan teknik-teknik statistika deskriptif dan pivoting tabel untuk menyelesaikan berbagai permasalahan studi kasus pada dataset.*

## 1. Statistika Deskriptif (Distribusi Uniform, Distribusi Normal, Skewness, dan Kurtosis)

### a. Distribusi Uniform

#### ▪ Import Library dan Load Dataset

```
1 import pandas as pd
2
3 ph = pd.read_csv("ph.csv", sep=';', decimal=',')
4 ph.head()
```

#### ▪ Mencari Nilai a dan b

```
1 data_ph = ph["x"]
2
3 mean = sum(data_ph) / len(data_ph)
4 print("Mean Dataset ph:", mean)
5
6 varians = sum((i - mean) ** 2 for i in data_ph) / len(data_ph)
7 std_dev = varians ** 0.5
8 print("Standar Deviasi ph:", std_dev)
```

```
Mean Dataset ph: 6.214999999999999
Standar Deviasi ph: 2.798709524048539
```

Kode diatas digunakan untuk mencari nilai a dan b dengan mencari nilai mean dan standar deviasi dalam dataset

#### ▪ Menggunakan persamaan mean dan standar deviasi untuk mendapatkan nilai a dan b sesuai dengan sifat dasar statistika dalam distribusi uniform.

```
1 a_plus_b = 2 * mean
2 print("Nilai a + b Dari ph Dataset :", a_plus_b)
3
4 b_min_a = std_dev * (12 ** 0.5)
5 print("Nilai b - a Dari ph Dataset :", b_min_a)
6
7 a = (a_plus_b - b_min_a) / 2
8 print("Nilai a Dari Distribusi Uniform :", a)
9
10 b = (a_plus_b + b_min_a) / 2
11 print("Nilai b Dari Distribusi Uniform :", b)
```

```
Nilai a + b Dari ph Dataset : 12.429999999999998
Nilai b - a Dari ph Dataset : 9.69501418255796
Nilai a Dari Distribusi Uniform : 1.3674929087210188
Nilai b Dari Distribusi Uniform : 11.06250709127898
```

Kode diatas menghitung dua parameter, "a" dan "b," dalam distribusi uniform berdasarkan mean dan standar deviasi dari dataset "ph." Parameter "a" adalah setengah dari selisih antara "a\_plus\_b" dan "b\_min\_a," sementara "b" adalah setengah dari jumlah keduanya. Hasilnya adalah dua parameter yang mendefinisikan distribusi uniform dengan rentang yang sesuai

dengan data dalam dataset "ph."

- Hitung probabilitas distribusi uniform dengan nilai a dan b

```
1 #nilai x1 dan x2
2 x1 = a
3 x2 = 5
4
5 # probabilitas uniform x < 5
6 prob_uni = (1 / (b - a)) * (x2 - a)
7 print("Probabilitas Uniform Dari Ph Dataset Dengan x < 5 Sebesar : ", prob_uni)
8
9 # presentase probablitas uniform x < 5
10 percent_prob = prob_uni * 100
11 print("Presentase Probabilitas Uniform Ph Dataset Dengan x < 5 Sebesar : ", percent_prob, "%")
```

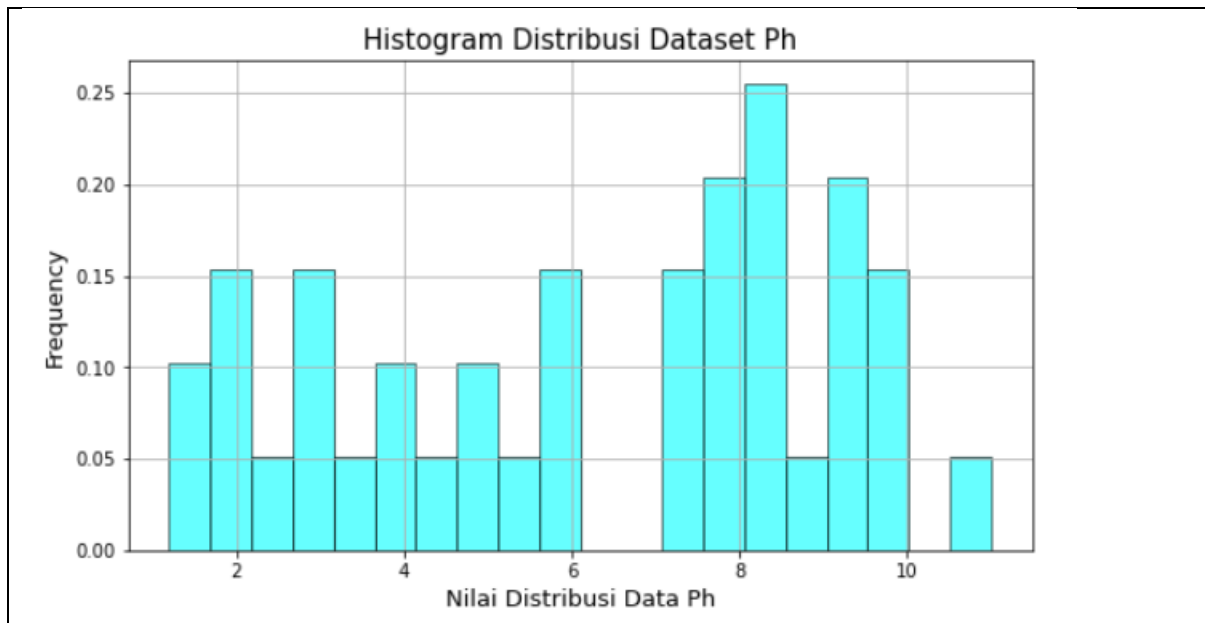
Probabilitas Uniform Dari Ph Dataset Dengan x < 5 Sebesar : 0.3746778522319366  
Presentase Probabilitas Uniform Ph Dataset Dengan x < 5 Sebesar : 37.46778522319366 %

Kode di atas menghitung probabilitas bahwa nilai acak "x" dalam distribusi uniform yang telah dihitung sebelumnya, yaitu antara "a" dan "b," kurang dari 5. Probabilitas ini dihitung dengan rumus probabilitas distribusi uniform, di mana kita menghitung panjang interval antara "a" dan 5, kemudian membaginya dengan panjang total interval antara "a" dan "b." Hasil probabilitas ini kemudian diubah menjadi persentase dengan mengalikannya dengan 100. Dengan demikian, kita mendapatkan probabilitas bahwa "x" kurang dari 5 dalam distribusi uniform dari dataset "ph," yang diungkapkan dalam bentuk persentase.

- Visualisasi Dataset

```
1 import matplotlib.pyplot as plt
2
3 #visualisasi histogram dataset ph.csv
4 data_ph = ph["x"]
5
6 plt.figure(figsize = (10, 6))
7 plt.hist(data_ph, bins = 20, density = True, color = "pink", alpha = 0.6,
8 plt.title("Histogram Distribusi Dataset Ph", size = 15, color = "black")
9 plt.xlabel("Nilai Distribusi Data Ph", size = 13, color = "black")
10 plt.ylabel("Frequency", size = 13, color = "black")
11 plt.grid(True)
12
13 plt.show()
```

Ditampilkan histogram:



Data ph berdistribusi uniform karena histogram berbentuk persegi panjang dengan tinggi yang sama pada setiap kelas.

#### b. Distribusi Normal

- Import Library dan Load Dataset

```
1 import pandas as pd
2
3 normal_data = pd.read_csv("normal_distribution.csv", sep=';', decimal=',')
4 normal_data
```

- Hitung Persamaan  $f(x)$  dan Gunakan Persamaan Distribusi Normal

```
1 # perhitungan f(x) menggunakan x di dataset normal
2 column = normal_data["x"]
3
4 # hitung mean
5 mean_normal = sum(column) / len(column)
6 print("Nilai Mean Dari Normal Distribution Dataset: {:.3f}".format(mean_normal))
7
8 # hitung standar deviasi
9 variance_normal = sum((i - mean_normal) ** 2 for i in column) / len(column)
10 std_dev_normal = variance_normal ** 0.5
11 print("Nilai Standar Deviasi Dari Normal Distribution Dataset: {:.3f}".format(std_dev_normal))
12
13 # hitung f(x)
14 def normal_distribution(x, mean, std_dev):
15     eksponen_normal = -(x - mean) ** 2 / (2 * std_dev ** 2)
16     persamaan_dist = (1 / (std_dev * ((2 * 3.141) ** 0.5))) * (2.718 ** eksponen_normal)
17     return persamaan_dist
18
19 # masuk nilai function
20 nilai_x = [normal_distribution(x, mean_normal, std_dev_normal) for x in column]
21 print("Hasil Persamaan f(x) Dari Nilai x Yang Terdapat Pada Normal Distribution Dataset:")
22
23 # menampilkan hasil persamaan f(x) dari x yang terdapat di normal distribution dataset
24 for i in range(len(column)):
25     x = column[i]
26     fx = nilai_x[i]
27     print("Nilai x: {:<6.2f} | Persamaan f(x): {:.5f}|".format(x, fx))
```

Kode ini digunakan untuk menghitung fungsi distribusi normal ( $f(x)$ ) berdasarkan nilai  $x$  dalam dataset distribusi normal. Pertama, kode menghitung nilai rata-rata (mean) dan standar

deviasi (standard deviation) dari dataset normal. Kemudian, fungsi `normal_distribution(x, mean, std_dev)` digunakan untuk menghitung  $f(x)$  berdasarkan rumus distribusi normal. Hasil  $f(x)$  kemudian dihitung untuk setiap nilai  $x$  dalam dataset normal. Hasilnya, kode ini menghasilkan  $f(x)$  untuk setiap nilai  $x$  dalam dataset distribusi normal, dan ini digunakan untuk memahami distribusi data dan nilai fungsinya pada berbagai titik. Output:

```

Nilai Mean Dari Normal Distribution Dataset: 0.079
Nilai Standar Deviasi Dari Normal Distribution Dataset: 2.294
Hasil Persamaan f(x) Dari Nilai x Yang Terdapat Pada Normal Distribution Dataset:
|Nilai x: -5.84 | Persamaan f(x): 0.00624|
|Nilai x: -4.86 | Persamaan f(x): 0.01715|
|Nilai x: -3.96 | Persamaan f(x): 0.03694|
|Nilai x: -3.74 | Persamaan f(x): 0.04353|
|Nilai x: -2.86 | Persamaan f(x): 0.07657|
|Nilai x: -2.35 | Persamaan f(x): 0.09931|
|Nilai x: -2.25 | Persamaan f(x): 0.10390|
|Nilai x: -2.16 | Persamaan f(x): 0.10804|
|Nilai x: -1.80 | Persamaan f(x): 0.12437|
|Nilai x: -1.69 | Persamaan f(x): 0.12920|
|Nilai x: -1.53 | Persamaan f(x): 0.13601|
|Nilai x: -1.52 | Persamaan f(x): 0.13642|
|Nilai x: -1.17 | Persamaan f(x): 0.14997|
|Nilai x: -1.17 | Persamaan f(x): 0.14997|
|Nilai x: -1.12 | Persamaan f(x): 0.15172|
|Nilai x: -0.94 | Persamaan f(x): 0.15758|

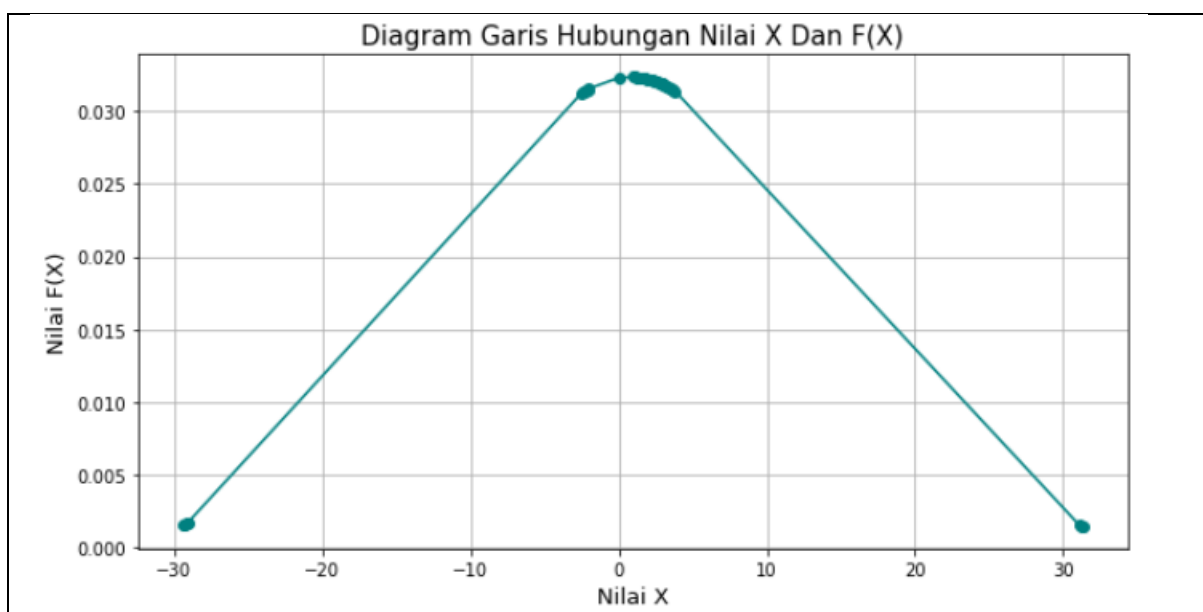
```

- Visualisasi Antara Nilai  $x$  dan  $f(x)$

```

1 import matplotlib.pyplot as plt
2
3 plt.figure(figsize = (9, 5))
4 plt.plot(column, nilai_x, color = "teal", marker = "o", linestyle = "-")
5 plt.title("Diagram Garis Hubungan Nilai X Dan F(X)", fontsize = 15)
6 plt.xlabel("Nilai X", fontsize = 13, color = "black")
7 plt.ylabel("Nilai F(X)", fontsize = 13, color = "black")
8 plt.tight_layout()
9 plt.grid(True)
10
11 plt.show()

```



Dari diagram garis tersebut, terlihat bahwa data memiliki distribusi normal, karena kurva memiliki bentuk lonceng

### c. Skewness

- Import Library dan Load Dataset

```
1 import pandas as pd
2
3 skewness = pd.read_csv("skewness.csv", sep=';', decimal=',')
4 skewness
```

- Hitung persamaan  $f(x)$  dari nilai  $x$  berdasarkan distribusi normal dan gunakan persamaan skewness

```
1 # perhitungan f(x) menggunakan x di dataset normal
2 column = skewness["x"]
3
4 # hitung mean
5 mean_normal = sum(column) / len(column)
6 print("Mean Normal Distribution Dataset: {:.3f}".format(mean_normal))
7
8 # hitung standar deviasi
9 variance_normal = sum((i - mean_normal) ** 2 for i in column) / len(column)
10 std_dev_normal = variance_normal ** 0.5
11 print("Standar Deviasi Normal Distribution Dataset: {:.3f}".format(std_dev_normal))
12
13 # hitung f(x)
14 def normal_distribution(x, mean, std_dev):
15     eksponen_normal = -(x - mean) ** 2 / (2 * std_dev ** 2)
16     persamaan_dist = (1 / (std_dev * ((2 * 3.141) ** 0.5))) * (2.718 ** eksponen_normal)
17     return persamaan_dist
18
19 # masuk nilai function
20 nilai_x = [normal_distribution(x, mean_normal, std_dev_normal) for x in column]
21 print("Persamaan f(x) Dari Nilai x Pada Normal Distribution Dataset:")
22
23 # penampilan hasil persamaan f(x) dari x yang terdapat di normal distribution dataset
24 for i in range(len(column)):
25     x = column[i]
26     fx = nilai_x[i]
27     print("|Nilai x: {:.6.2f} | Persamaan f(x): {:.5f}|".format(x, fx))
28
29 # Hitung skewness
30 median_normal = column.median() # Untuk menghitung skewness, kita gunakan mean
31 skewness = (3 * (mean_normal - median_normal)) / std_dev_normal
32 print("Skewness dari dataset adalah:", skewness)
```

Kode ini digunakan untuk menghitung dan mengevaluasi fungsi distribusi normal ( $f(x)$ ) berdasarkan nilai  $x$  dalam dataset yang memiliki distribusi skewness. Terlebih dahulu, kode menghitung rata-rata (mean) dan standar deviasi (standard deviation) dari dataset skewness. Kemudian, fungsi `normal_distribution(x, mean, std_dev)` digunakan untuk menghitung  $f(x)$  berdasarkan rumus distribusi normal. Hasil  $f(x)$  kemudian dihitung untuk setiap nilai  $x$  dalam dataset skewness. Selain itu, kode juga menghitung skewness dari dataset menggunakan rumus skewness yang menggambarkan kemiringan distribusi data. Hasilnya, kode ini menghasilkan  $f(x)$  untuk setiap nilai  $x$  dalam dataset skewness dan juga mengukur kemiringan distribusi data melalui skewness. Hal ini membantu dalam memahami sejauh mana distribusi data deviasi dari distribusi normal. Output:

Mean Normal Distribution Dataset: -0.031  
 Standar Deviasi Normal Distribution Dataset: 2.708  
 Persamaan  $f(x)$  Dari Nilai  $x$  Pada Normal Distribution Dataset:

Nilai $x$ : -10.00	Persamaan $f(x)$ : 0.00017
Nilai $x$ : -3.30	Persamaan $f(x)$ : 0.07110
Nilai $x$ : -3.20	Persamaan $f(x)$ : 0.07429
Nilai $x$ : -3.10	Persamaan $f(x)$ : 0.07752
Nilai $x$ : -3.00	Persamaan $f(x)$ : 0.08077
Nilai $x$ : -2.90	Persamaan $f(x)$ : 0.08405
Nilai $x$ : -2.80	Persamaan $f(x)$ : 0.08735
Nilai $x$ : -2.70	Persamaan $f(x)$ : 0.09065
Nilai $x$ : -2.60	Persamaan $f(x)$ : 0.09394
Nilai $x$ : -2.50	Persamaan $f(x)$ : 0.09722
Nilai $x$ : -2.40	Persamaan $f(x)$ : 0.10048
Nilai $x$ : -2.30	Persamaan $f(x)$ : 0.10371
Nilai $x$ : -2.20	Persamaan $f(x)$ : 0.10690
Nilai $x$ : -2.10	Persamaan $f(x)$ : 0.11003

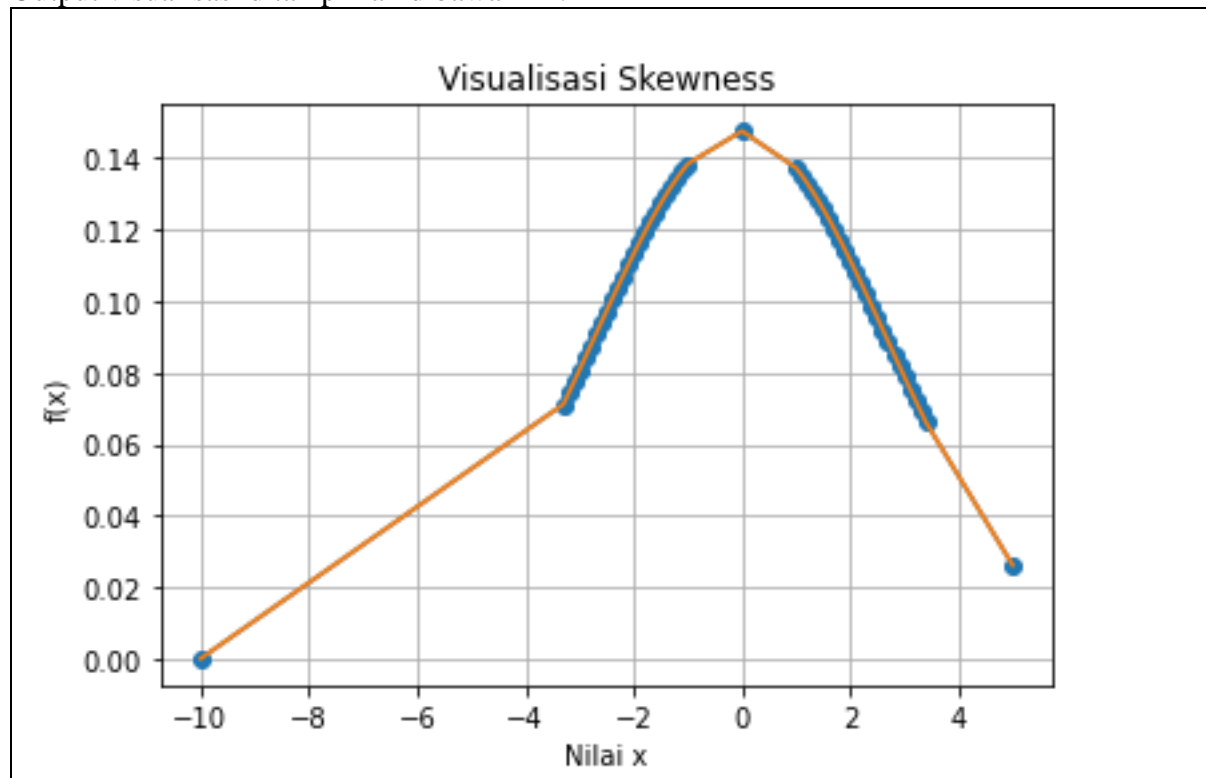
.

Skewness dari dataset adalah: -0.5879543070683356

#### Visualisasi Diagram Garis

```
1 plt.plot(column, nilai_x, marker='o', linestyle='--')
2 plt.plot(column, nilai_x)
3 plt.xlabel("Nilai x")
4 plt.ylabel("f(x)")
5 plt.title("Visualisasi Distribusi Normal")
6 plt.grid(True)
7 plt.show()
```

Output visualisasi ditampilkan dibawah ini:



Berdasarkan histogram distribusi data ph di atas, dapat disimpulkan bahwa data tersebut

berdistribusi positif skew. Hal ini terlihat dari bentuk histogram yang memanjang ke arah kanan. Artinya, nilai data Ph yang lebih besar dari rata-rata lebih sering muncul daripada nilai data ph yang lebih kecil dari rata-rata.

#### d. Kurtosis

- Import Library dan Load Dataset

```
1 import pandas as pd
2
3 kurtosis = pd.read_csv("kurtosis.csv", sep=';', decimal=',')
4 kurtosis
```

- Hitung persamaan  $f(x)$  dari nilai  $x$  berdasarkan distribusi normal dan gunakan persamaan kurtosis

```
1 import numpy as np
2
3 # perhitungan f(x) menggunakan x di dataset kurtosis
4 column = kurtosis["x"]
5
6 # hitung mean
7 mean_normal = sum(column) / len(column)
8 print("Mean Dataset Kurtosis: {:.3f}".format(mean_normal))
9
10 # hitung standar deviasi
11 variance_normal = sum((i - mean_normal) ** 2 for i in column) / len(column)
12 std_dev_normal = variance_normal ** 0.5
13 print("Standar Deviasi Dataset Kurtosis: {:.3f}".format(std_dev_normal))
14
15 # hitung f(x)
16 def normal_distribution(x, mean, std_dev):
17     eksponen_normal = -(x - mean) ** 2 / (2 * std_dev ** 2)
18     persamaan_dist = (1 / (std_dev * ((2 * np.pi) ** 0.5))) * (np.exp(eksponen_normal))
19     return persamaan_dist
20
21 # masukkan nilai ke dalam fungsi
22 nilai_x = [normal_distribution(x, mean_normal, std_dev_normal) for x in column]
23 print("Hasil Persamaan f(x) Dari Nilai x Yang Terdapat Pada Dataset Kurtosis :")
24
25 # tampilkan hasil persamaan f(x) dari x yang terdapat di dataset kurtosis
26 for i in range(len(column)):
27     x = column[i]
28     fx = nilai_x[i]
29     print("|Nilai x: {:<6.2f} | Persamaan f(x): {:.5f}|".format(x, fx))
30
31 # Hitung kurtosis
32 kurtosis_value = sum((x - mean_normal) ** 4 for x in column) / (len(column) * (std_dev_normal ** 4))
33 print("Kurtosis dari dataset adalah:", kurtosis_value)
```

Kode ini digunakan untuk menghitung dan mengevaluasi fungsi distribusi normal ( $f(x)$ ) berdasarkan nilai  $x$  dalam dataset kurtosis. Pertama, rata-rata (mean) dan standar deviasi (standard deviation) dari dataset dihitung. Kemudian, fungsi distribusi normal diterapkan pada setiap nilai dalam dataset untuk menghitung  $f(x)$ . Selain itu, kode juga menghitung kurtosis dari dataset, yang menggambarkan karakteristik keruncingan atau datar distribusi data. Nilai kurtosis yang lebih besar dari 3 menunjukkan distribusi lebih kerucut (leptokurtik), sementara nilai yang lebih kecil menunjukkan distribusi lebih datar (platykurtik) dibandingkan dengan distribusi normal (kurtosis normal adalah 3). Hal ini membantu dalam memahami karakteristik distribusi data dalam konteks kurtosis. Output:



```

Mean Dataset Kurtosis: 0.760
Standar Deviasi Dataset Kurtosis: 12.329
Hasil Persamaan f(x) Dari Nilai x Yang Terdapat Pada Dataset Kurtosis :
|Nilai x: -29.40 | Persamaan f(x): 0.00162|
|Nilai x: -29.30 | Persamaan f(x): 0.00166|
|Nilai x: -29.20 | Persamaan f(x): 0.00169|
|Nilai x: -29.10 | Persamaan f(x): 0.00172|
|Nilai x: -2.50  | Persamaan f(x): 0.03125|
|Nilai x: -2.40  | Persamaan f(x): 0.03131|
|Nilai x: -2.30  | Persamaan f(x): 0.03138|
|Nilai x: -2.20  | Persamaan f(x): 0.03144|
|Nilai x: -2.10  | Persamaan f(x): 0.03150|
|Nilai x: -2.00  | Persamaan f(x): 0.03156|
|Nilai x: 0.00   | Persamaan f(x): 0.03230|
|Nilai x: 1.00   | Persamaan f(x): 0.03235|
|Nilai x: 1.10   | Persamaan f(x): 0.03235|
|Nilai x: 1.20   | Persamaan f(x): 0.03234|
|Nilai x: 1.30   | Persamaan f(x): 0.03233|
.
.
Kurtosis dari dataset adalah: 5.860396932336173

```

- Distribusi data mengarah ke platykurtic, mesokurtic, atau leptokurtic?

```

1 # Mengambil nilai kurtosis
2 kurtosis = kurtosis_value
3
4 if kurtosis < 3:
5     decision = "Platykurtic (ekor tipis)"
6     explanation = "Distribusi data memiliki kurtosis kurang dari 3, yang berarti ekornya sangat tipis dibandingkan dengan d
7 elif kurtosis == 3:
8     decision = "Mesokurtic"
9     explanation = "Distribusi data memiliki kurtosis sama dengan 3, yang menunjukkan bahwa distribusi ini mirip dengan d
10 else:
11     decision = "Leptokurtic (ekor tebal)"
12     explanation = "Distribusi data memiliki kurtosis lebih dari 3, yang menunjukkan distribusi ini menghasilkan lebih bar
13
14 print("Distribusi data mengarah ke:", decision)
15 print("Penjelasan:", explanation)

```

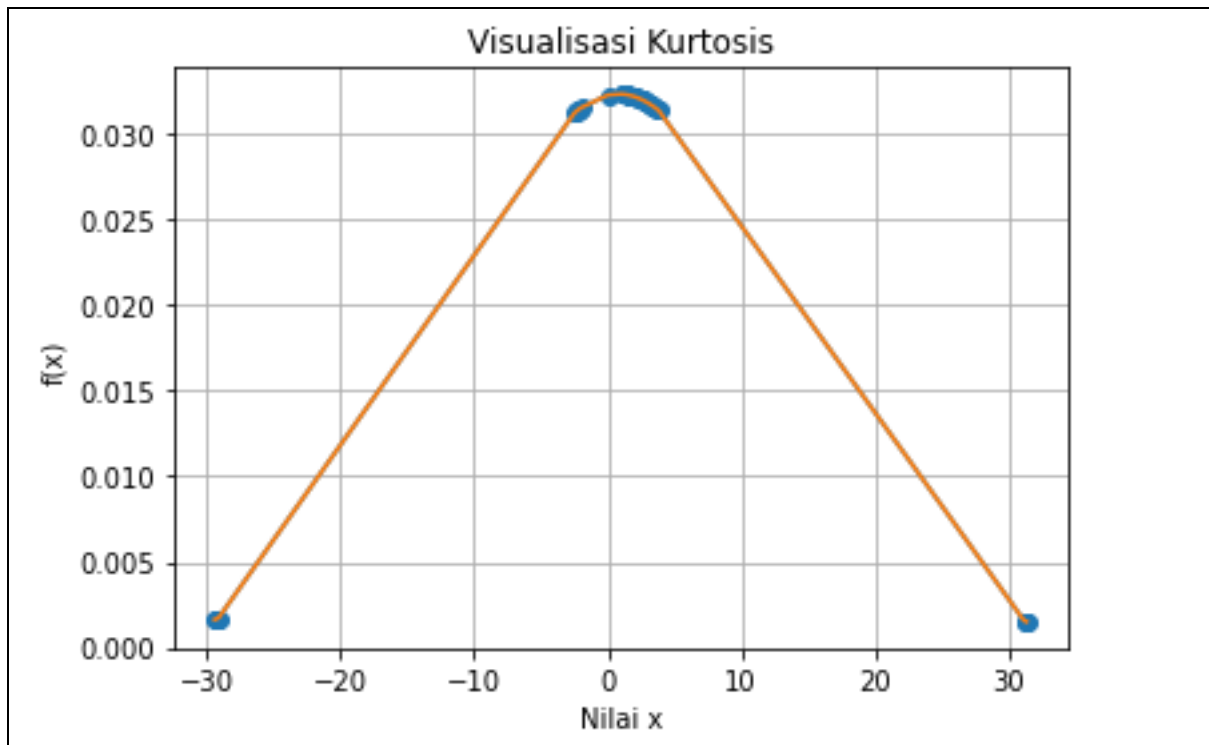
Distribusi data mengarah ke: Leptokurtic (ekor tebal)  
 Penjelasan: Distribusi data memiliki kurtosis lebih dari 3, yang menunjukkan distribusi ini menghasilkan lebih banyak outlier.

- Visualisasi Diagram Garis

```

1 plt.plot(column, nilai_x, marker='o', linestyle='-')
2 plt.plot(column, nilai_x)
3 plt.xlabel("Nilai x")
4 plt.ylabel("f(x)")
5 plt.title("Visualisasi Kurtosis")
6 plt.grid(True)
7 plt.show()

```



## 2. Grouping Dataset: Pivoting Table

Lakukanlah pivoting tabel menggunakan kode *script* pemrograman Python pembelian bahan pokok makanan yang dikirimkan oleh supplier pada kuartal tertentu menjadi tabel seperti berikut:

### a. Pivoting Tabel Berdasarkan Jumlah Pembelian Bahan Pokok (kg) per Kuartal

Row Labels	2015	2016	2017	Grand Total
<b>Bawang Merah</b>	<b>1450</b>	<b>1278</b>	<b>1574</b>	<b>4302</b>
Kuartal 1	406	296	408	1110
Kuartal 2	277	248	427	952
Kuartal 3	431	403	430	1264
Kuartal 4	336	331	309	976
<b>Bawang Putih</b>	<b>1419</b>	<b>1659</b>	<b>1601</b>	<b>4679</b>
Kuartal 1	313	534	355	1202
Kuartal 2	407	447	443	1297
Kuartal 3	361	332	506	1199
Kuartal 4	338	346	297	981
<b>Cabai</b>	<b>1543</b>	<b>1568</b>	<b>1612</b>	<b>4723</b>
Kuartal 1	309	272	221	802
Kuartal 2	364	473	431	1268
Kuartal 3	519	366	453	1338
Kuartal 4	351	457	507	1315
...	...	...	...	...

- Import Library dan Load Dataset

```
1 import pandas as pd
2
3 bp = pd.read_csv('data_bahan_pokok.csv', sep=';')
4 bp
```

	TANGGAL	BARANG	SUPLIER	BULAN	TAHUN	KUARTAL	JUMLAH	HARGA	TOTAL
0	01/01/15	Cabai	Bejo	January	2015	Kuartal 1	55	IDR30.550	IDR1.680.250
1	06/01/15	Kunyit	Bejo	January	2015	Kuartal 1	72	IDR12.964	IDR933.408
2	11/01/15	Bawang Merah	Painah	January	2015	Kuartal 1	81	IDR28.875	IDR2.338.875
3	18/01/15	Bawang Putih	Tarno	January	2015	Kuartal 1	146	IDR20.812	IDR3.038.552
4	22/01/15	Kencur	Bejo	January	2015	Kuartal 1	159	IDR9.785	IDR1.555.815
...	...	...	...	...	...	...	...	...	...
211	07/12/17	Kunyit	Bejo	December	2017	Kuartal 4	77	IDR13.604	IDR1.047.508
212	12/12/17	Bawang Merah	Painah	December	2017	Kuartal 4	104	IDR24.709	IDR2.569.736
213	18/12/17	Bawang Putih	Tarno	December	2017	Kuartal 4	124	IDR22.074	IDR2.737.176
214	22/12/17	Kencur	Bejo	December	2017	Kuartal 4	128	IDR8.965	IDR1.147.520
215	26/12/17	Jahe	Painah	December	2017	Kuartal 4	191	IDR22.493	IDR4.296.163

- Pivoting Tabel

```
1 def create_pivot_for_item(item_name, bp):
2     filtered_data = bp[bp['BARANG'] == item_name]
3     pivot = pd.pivot_table(filtered_data, values='JUMLAH', index='KUARTAL', columns='TAHUN', aggfunc='sum', fill_value=0,
4                             margins=True, margins_name='Grand Total')
5     pivot = pivot[[2015, 2016, 2017, 'Grand Total']].loc[['Kuartal 1', 'Kuartal 2', 'Kuartal 3', 'Kuartal 4', 'Grand Total']]
6     pivot.loc[item_name] = pivot.loc['Grand Total']
7     return pivot.loc[[item_name, 'Kuartal 1', 'Kuartal 2', 'Kuartal 3', 'Kuartal 4']]
8
9 unique_items = bp['BARANG'].unique()
10
11 sorted_items = sorted(unique_items)
12 sorted_pivot = pd.concat([create_pivot_for_item(item, bp) for item in sorted_items])
13 sorted_pivot.rename_axis("Row Labels", axis=1, inplace=True)
14 sorted_pivot
```

Kode diatas digunakan untuk menghasilkan tabel pivot dari dataset bahan pokok menggunakan python dan library pandas. Tujuannya untuk merangkum dan mengorganisir data pembelian bahan pokok berdasarkan jenis barang, kuartal, dan tahun. Kode tersebut membantu dalam mengorganisir data pembelian bahan pokok dengan melakukan pengelompokan, menghitung total pembelian, dan menyesuaikan label. Ini bertujuan untuk menyajikan data pembelian bahan pokok dengan tampilan yang lebih terstruktur dan mudah dipahami.

▪ Hasil Pivoting

Row Labels	2015	2016	2017	Grand Total
<b>KUARTAL</b>				
<b>Bawang Merah</b>	1450	1278	1574	4302
<b>Kuartal 1</b>	406	296	408	1110
<b>Kuartal 2</b>	277	248	427	952
<b>Kuartal 3</b>	431	403	430	1264
<b>Kuartal 4</b>	336	331	309	976
<b>Bawang Putih</b>	1419	1659	1601	4679
<b>Kuartal 1</b>	313	534	355	1202
<b>Kuartal 2</b>	407	447	443	1297
<b>Kuartal 3</b>	361	332	506	1199
<b>Kuartal 4</b>	338	346	297	981
<b>Cabai</b>	1543	1568	1612	4723
<b>Kuartal 1</b>	309	272	221	802
<b>Kuartal 2</b>	364	473	431	1268
<b>Kuartal 3</b>	519	366	453	1338
<b>Kuartal 4</b>	351	457	507	1315
<b>Jahe</b>	1317	1677	1659	4653
<b>Kuartal 1</b>	276	515	435	1226
<b>Kuartal 2</b>	251	262	387	900
<b>Kuartal 3</b>	363	488	426	1277
<b>Kuartal 4</b>	427	412	411	1250
<b>Kencur</b>	1607	1489	1376	4472
<b>Kuartal 1</b>	527	376	340	1243
<b>Kuartal 2</b>	345	328	420	1093
<b>Kuartal 3</b>	370	470	269	1109
<b>Kuartal 4</b>	365	315	347	1027
<b>Kunyit</b>	1640	1446	1419	4505
<b>Kuartal 1</b>	374	284	398	1056
<b>Kuartal 2</b>	439	345	255	1039
<b>Kuartal 3</b>	355	368	350	1073
<b>Kuartal 4</b>	472	449	416	1337

b. Pivoting Tabel Berdasarkan Total Pembelian Bahan Pokok (Rupiah) per Kuartal

Row Labels	2015	2016	2017	Grand Total
<b>Bawang Merah</b>	<b>37823531</b>	<b>31731767</b>	<b>40941357</b>	<b>110496655</b>
Kuartal 1	11037115	6820091	10456137	28313343
Kuartal 2	7596610	6251504	11075213	24923327
Kuartal 3	10800386	10587768	11697016	33085170
Kuartal 4	8389420	8072404	7712991	24174815
<b>Bawang Putih</b>	<b>31078455</b>	<b>37670606</b>	<b>35263522</b>	<b>104012583</b>
Kuartal 1	6812135	12662576	7962251	27436962
Kuartal 2	8932531	10457967	9784401	29174899
Kuartal 3	8082531	7110937	10945924	26139392
Kuartal 4	7251258	7439126	6570946	21261330
<b>Cabai</b>	<b>45057879</b>	<b>44441269</b>	<b>46168221</b>	<b>135667369</b>
Kuartal 1	9219854	7652138	5787840	22659832
Kuartal 2	10617522	13099592	12230465	35947579
Kuartal 3	15774604	9718410	14312881	39805895
Kuartal 4	9445899	13971129	13837035	37254063
...	...	...	...	...

## ■ Pivoting Tabel

```

1 def create_pivot_for_item(item_name, bp):
2
3     bp['TOTAL'] = bp['TOTAL'].astype(str).str.replace('IDR', '', regex=False).str.replace('.', '', regex=False).astype(int)
4
5     filtered_data = bp[bp['BARANG'] == item_name]
6     pivot = pd.pivot_table(filtered_data, values='TOTAL', index='KUARTAL', columns='TAHUN', aggfunc='sum',
7                             fill_value=0, margins=True, margins_name='Grand Total')
8     pivot = pivot[[2015, 2016, 2017, 'Grand Total']].loc[['Kuartal 1', 'Kuartal 2', 'Kuartal 3', 'Kuartal 4',
9                                                         'Grand Total']]
10    pivot.loc[item_name] = pivot.loc['Grand Total']
11    return pivot.loc[[item_name, 'Kuartal 1', 'Kuartal 2', 'Kuartal 3', 'Kuartal 4']]
12
13 unique_items = bp['BARANG'].unique()
14
15 sorted_items = sorted(unique_items)
16 sorted_pivot = pd.concat([create_pivot_for_item(item, bp) for item in sorted_items])
17 sorted_pivot.rename_axis("Row Labels", axis=1, inplace=True)
18 sorted_pivot

```

Penjelasan Kode ini digunakan untuk membuat pivot table yang merangkum data penjualan item-item tertentu dari suatu dataset. Prosesnya dimulai dengan mengonversi kolom "TOTAL" ke dalam format yang sesuai dan menghapus simbol mata uang IDR serta tanda titik. Kemudian, data yang sudah difilter berdasarkan nama item yang disediakan, diproses untuk menciptakan pivot table dengan total penjualan per kuartal dan total tahun. Pivot table ini kemudian difilter kembali untuk hanya menampilkan kuartal dan total tahun tertentu, serta disusun sesuai urutan. Hasil akhirnya adalah pivot table yang menyajikan total penjualan item-item yang berbeda per kuartal dan tahun, disertai dengan grand total, dengan nama item yang diurutkan secara alfabetis.

## ■ Hasil Pivoting

Row Labels	2015	2016	2017	Grand Total					
<b>KUARTAL</b>									
<b>Bawang Merah</b>	37823531	31731767	40941357	110496655	<b>Jahe</b>	29984636	36550966	38324202	104859804
<b>Kuartal 1</b>	11037115	6820091	10456137	28313343	<b>Kuartal 1</b>	6204072	11010367	10148692	27363131
<b>Kuartal 2</b>	7596610	6251504	11075213	24923327	<b>Kuartal 2</b>	5775616	5880108	8595257	20250981
<b>Kuartal 3</b>	10800386	10587768	11697016	33085170	<b>Kuartal 3</b>	7806801	10696194	10448307	28951302
<b>Kuartal 4</b>	8389420	8072404	7712991	24174815	<b>Kuartal 4</b>	10198147	8964297	9131946	28294390
<b>Bawang Putih</b>	31078455	37670606	35263522	104012583	<b>Kencur</b>	15030665	13539469	12267295	40837429
<b>Kuartal 1</b>	6812135	12662576	7962251	27436962	<b>Kuartal 1</b>	4981422	3332834	2922221	11236477
<b>Kuartal 2</b>	8932531	10457967	9784401	29174899	<b>Kuartal 2</b>	3362034	3039520	3769896	10171450
<b>Kuartal 3</b>	8082531	7110937	10945924	26139392	<b>Kuartal 3</b>	3173993	4161334	2297363	9632690
<b>Kuartal 4</b>	7251258	7439126	6570946	21261330	<b>Kuartal 4</b>	3513216	3005781	3277815	9796812
<b>Cabai</b>	45057879	44441269	46168221	135667369	<b>Kunyit</b>	19931524	18179929	18339347	56450800
<b>Kuartal 1</b>	9219854	7652138	5787840	22659832	<b>Kuartal 1</b>	4048734	3119536	4489786	11658056
<b>Kuartal 2</b>	10617522	13099592	12230465	35947579	<b>Kuartal 2</b>	5524089	4272801	3401235	13198125
<b>Kuartal 3</b>	15774604	9718410	14312881	39805895	<b>Kuartal 3</b>	4177633	5073847	4809108	14060588
<b>Kuartal 4</b>	9445899	13971129	13837035	37254063	<b>Kuartal 4</b>	6181068	5713745	5639218	17534031