

LAPORAN EVALUASI TENGAH SEMESTER STATISTIK KOMPUTASI



Oleh:

Syahrial Rizky [22081010020]

Dosen Pengampu:

Yisti Vita Via S., ST., M. Kom.

**FAKULTAS ILMU KOMPUTER
UPN “VETERAN” JAWA TIMUR
SURABAYA 2023**

I. PENGOLAHAN DATA

A. Dasar Teori

Pengolahan data dalam statistik komputasi mencakup konsep – konsep inti seperti mean, modus, median, standar deviasi, dan varians. Mean atau rata – rata merupakan salah satu ukuran tendensi sentral yang digunakan dalam statistika. Modus merupakan nilai – nilai dalam suatu set data yang paling sering muncul atau memiliki frekuensi tertinggi. Modus memberikan indikasi tentang nilai yang paling umum atau dominan dalam kumpulan suatu set data. Median adalah nilai tengah dalam suatu set data ketika nilai – nilai diurutkan dalam urutan numerik. Standar deviasi merupakan ukuran seberapa tersebar atau seberapa jauh nilai – nilai dalam suatu data set dari nilai rata rata nya. Standar deviasi memberikan informasi tentang variabilitas atau dispersi data. Varians adalah ukuran statistik yang mengukur seberapa jauh nilai – nilai dalam suatu set data tersebar dari nilai rata - rata nya. Dengan pemahaman yang mendalam tentang konsep – konsep diatas diharap pengolahan data statistik komputasi menjadi lebih efisien dan mudah untuk dipahami.

B. Kode Program

```
import pandas as pd
import numpy as np
from statistics import mode, StatisticsError

# Membaca data dari file Excel
path = r"C:\Users\Lenovo\Desktop\statkomMeet\statkom\data_syahrial.xlsx"

df = pd.read_excel(path, skiprows=0, usecols='A:E', nrows=2120)

# Menghitung mean, median, modus, standar deviasi, dan varians untuk setiap kolom
for kolom in df.columns:

    # print(df.columns)
    data = df[kolom].tolist()

    # Menghitung mean
    mean = np.mean(data)

    # Menghitung median
    median = np.median(data)
```

```

# Menghitung modus
try:
    modus = mode(data)
except StatisticsError:
    modus = "Tidak ada modus"

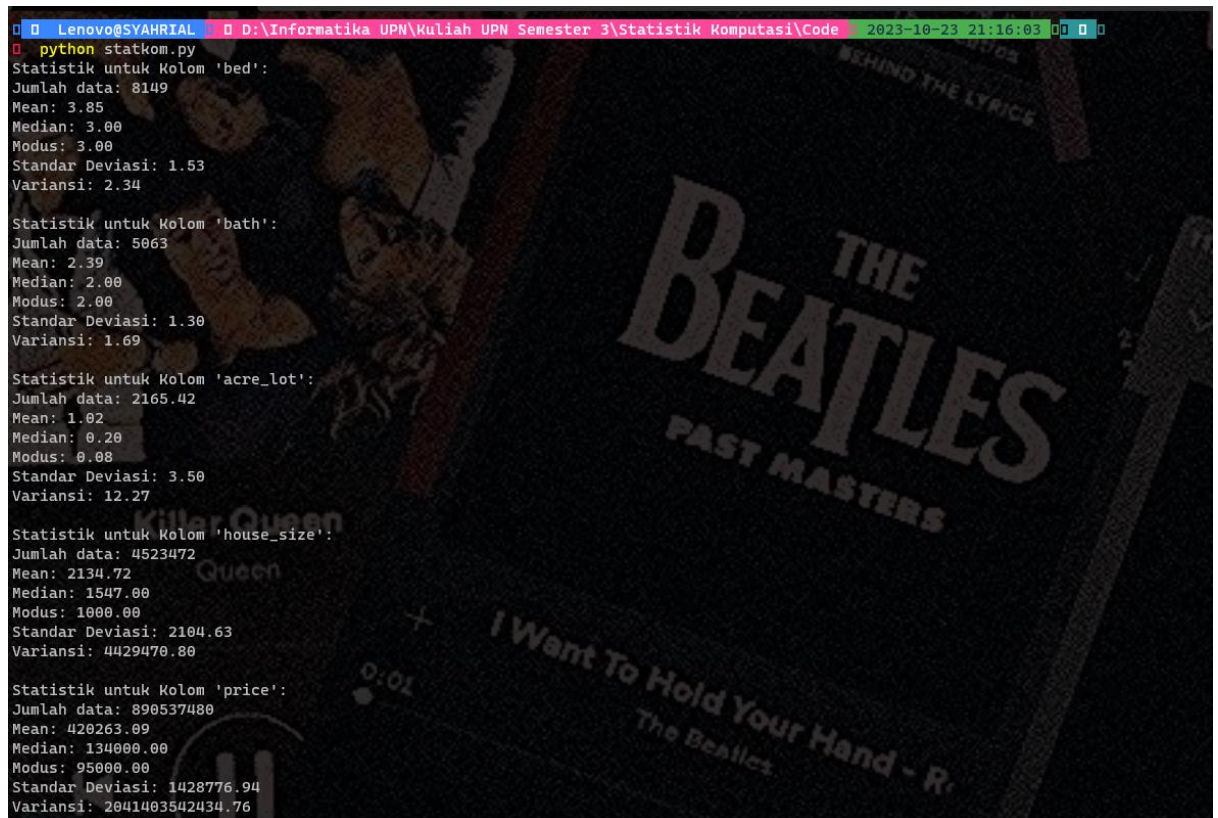
# Menghitung standar deviasi
std_deviation = round(np.std(data), 2)

# Menghitung varians
variance = round(np.var(data), 2)

# Menampilkan hasil statistik
print(f"Statistik untuk Kolom '{kolom}':")
print(f'Jumlah data: {np.sum(data)}')
print(f"Mean: {mean:.2f}")
print(f"Median: {median:.2f}")
print(f"Modus: {modus:.2f}")
print(f"Standar Deviasi: {std_deviation:.2f}")
print(f"Variansi: {variance:.2f}")
print()

```

C. Hasil Running Program



```

C:\Users\Lenovo\SYAHRIAL> python statkom.py
Statistik untuk Kolom 'bed':
Jumlah data: 8149
Mean: 3.85
Median: 3.00
Modus: 3.00
Standar Deviasi: 1.53
Variansi: 2.34

Statistik untuk Kolom 'bath':
Jumlah data: 5063
Mean: 2.39
Median: 2.00
Modus: 2.00
Standar Deviasi: 1.30
Variansi: 1.69

Statistik untuk Kolom 'acre_lot':
Jumlah data: 2165.42
Mean: 1.02
Median: 0.20
Modus: 0.08
Standar Deviasi: 3.50
Variansi: 12.27

Statistik untuk Kolom 'house_size':
Jumlah data: 4523472
Mean: 2134.72
Median: 1547.00
Modus: 1000.00
Standar Deviasi: 2104.63
Variansi: 4429470.80

Statistik untuk Kolom 'price':
Jumlah data: 890537480
Mean: 420263.09
Median: 134000.00
Modus: 95000.00
Standar Deviasi: 1428776.94
Variansi: 2041403542434.76

```

D. Analisis Hasil Running Program

Dari hasil running program pengolahan data menunjukkan bahwa program menampilkan hasil yang hampir serupa dengan yang ada pada Excel. Hanya terdapat sedikit perbedaan angka, hal tersebut mungkin disebabkan oleh pembulatan digit desimal yang ada pada program dan Excel

bed	bath	acre_lot	house_size	price	
8149	5063	2165.42	4523472	890537480	Jumlah
3.845682	2.389335	1.021907	2134.720151	420263.0864	Mean
3	2	0.08	1000	95000	Modus
3	2	0.2	1547	134000	Median
1.530079	1.300432	3.503913	2105.127584	1429114.193	Standard Deviasi
2.341142	1.691123	12.2774	4431562.145	2.04237E+12	Variansi

II. PENYAJIAN DATA

A. Dasar Teori

Penyajian data statistik komputasi mencakup prinsip – prinsip seperti Distribusi frekuensi, Distribusi frekuensi kumulatif (lebih dari dan kurang dari), Distribusi Relatif, dan Distribusi Relatif kumulatif (lebih dari dan kurang dari). Pengertian yang komprehensif terhadap konsep-konsep ini memainkan peran krusial dalam merangkum dan memvisualisasikan data secara terstruktur, yang pada gilirannya memungkinkan analisis mendalam dalam statistik komputasi. Dengan pemahaman yang kuat mengenai distribusi frekuensi dan variasinya, presentasi data dapat dihasilkan dengan lebih informatif dan relevan, khususnya dalam konteks statistik komputasi.

B. Kode Program

1. Normalisasi

```
import pandas as pd
import matplotlib.pyplot as plt
```

```

path = r"D:\Informatika UPN\Kuliah UPN Semester 3\Statistik
Komputasi\Code\excel\consume_data.xlsx"
df = pd.read_excel(path, skiprows=0, usecols='A:E', nrows=2120)

print(df.columns)
# Ganti dengan nama kolom yang sesuai
kolom_normalisasi = ['bed', 'bath', 'acre_lot', 'house_size', 'price']

# Normalisasi data
for kolom in kolom_normalisasi:
    # Menghitung nilai minimum dan maksimum dari setiap kolom
    min_val = df[kolom].min()
    max_val = df[kolom].max()

    # Min-Max Scaling
    df[kolom] = (df[kolom] - min_val) / (max_val - min_val)

# Menampilkan hasil normalisasi

print("Data setelah dinormalisasi:")
print(df.head(30))

```

2. Penyajian Data dan Grafik

```

import pandas as pd
import matplotlib.pyplot as plt

path = r"D:\Informatika UPN\Kuliah UPN Semester 3\Statistik
Komputasi\Code\excel\consume_data.xlsx"
df = pd.read_excel(path, skiprows=0, usecols='A:E', nrows=2120)

print(df.columns)
# Ganti dengan nama kolom yang sesuai
kolom_normalisasi = ['bed', 'bath', 'acre_lot', 'house_size', 'price']

# Normalisasi data
for kolom in kolom_normalisasi:
    # Menghitung nilai minimum dan maksimum dari setiap kolom
    min_val = df[kolom].min()
    max_val = df[kolom].max()

    # Min-Max Scaling
    df[kolom] = (df[kolom] - min_val) / (max_val - min_val)

# Menampilkan hasil normalisasi

print("Data setelah dinormalisasi:")
print(df.head(30))

```

```

# Mendefinisikan function
def kelompokDataBed():

    data = pd.read_excel(path, skiprows=0, usecols='A', nrows=2120)

    interval = [1, 4, 7, 10, 14]

    # Membuat DataFrame dari data
    data_kelompok = data.copy()

    # Mengelompokkan data ke dalam interval
    data_kelompok['Kelompok data bed'] = pd.cut(data_kelompok['bed'],
bins=interval, labels=[
                                                    '1-3', '4-7', '8-10', '11-
13'], right=False)

    # Menghitung frekuensi untuk setiap kelompok
    frekuensi = data_kelompok['Kelompok data bed'].value_counts().sort_index()

    # Mencari Distribusi Kumulatif Kurang dari
    distribusi_kurang_dari = [frekuensi.iloc[:i+1].sum()
                             for i in range(len(frekuensi))]

    # Mencari Distribusi Kumulatif Lebih dari
    distribusi_lebih_dari = [frekuensi.iloc[i:].sum()
                             for i in range(len(frekuensi))]

    # Mencari Distribusi Relatif
    distribusi_relatif = frekuensi / frekuensi.sum() * 100

    # Mencari Distribusi Relatif Kumulatif Kurang dari
    distribusi_relatif_kurang_dari = [
        sum(distribusi_relatif.iloc[:i+1]) for i in
range(len(distribusi_relatif))]

    # Mencari Distribusi Relatif Kumulatif Lebih dari
    distribusi_relatif_lebih_dari = [
        sum(distribusi_relatif.iloc[i:]) for i in
range(len(distribusi_relatif))]

    # Menampilkan hasil
    print("===== KELOMPOK DATA BED =====\n")
    print("Data dengan Kelompok dan Frekuensinya:")
    print(data_kelompok.head(100))
    print("\nFrekuensi untuk Setiap Kelompok:")
    print(frekuensi)
    print(f"\nTotal : {frekuensi.sum()} ")

```

```

print("\nDistribusi Kumulatif Kurang dari:")
print(list(zip(frekuensi.index, distribusi_kurang_dari)))
print("\nDistribusi Kumulatif Lebih dari:")
print(list(zip(frekuensi.index, distribusi_lebih_dari)))
print("\nDistribusi Relatif:")
print(list(zip(frekuensi.index, distribusi_relatif)))
print("\nDistribusi Relatif Kumulatif Kurang dari:")
print(list(zip(frekuensi.index, distribusi_relatif_kurang_dari)))
print("\nDistribusi Relatif Kumulatif Lebih dari:")
print(list(zip(frekuensi.index, distribusi_relatif_lebih_dari)))
print("=====\n")

# Membuat grafik Frekuensi kumulatif
grafikFrekKumulatif = plt.figure(figsize=(10, 6))
plt.bar(frekuensi.index, frekuensi.values, label='Frekuensi')
plt.plot(frekuensi.index, distribusi_kurang_dari, marker='o',
         linestyle='-', color='skyblue', label='Freq. Kumulatif kurang
dari')

plt.plot(frekuensi.index, distribusi_lebih_dari, marker='o',
         linestyle='-', color='lime', label='Freq. Kumulatif lebih dari')

# Menampilkan dan memberi nama
plt.xlabel('Frekuensi')
plt.ylabel('Nilai')
plt.title('Grafik Data Kelompok Bed')
plt.legend()
plt.grid(True)
grafikFrekKumulatif.show()

# Membuat grafik Frekuensi Relatif
plt.figure(figsize=(10, 6))
plt.plot(frekuensi.index, distribusi_relatif, marker='o',
         linestyle='-', color='red', label='Freq. Relatif')

plt.plot(frekuensi.index, distribusi_relatif_kurang_dari, marker='o',
         linestyle='-', color='orange', label='Freq. Relatif kurang dari')

plt.plot(frekuensi.index, distribusi_relatif_lebih_dari, marker='o',
         linestyle='-', color='yellow', label='Freq. Relatif lebih dari')

# Menampilkan dan memberi nama
plt.xlabel('Kelompok Data Bed (Relatif)')
plt.ylabel('Nilai')
plt.title('Grafik Data Kelompok Bed')
plt.legend()
plt.grid(True)

```



```
plt.show()

# Panggil function
kelompokDataBed()
print("\n")
```

C. Hasil Running Program

1. Hasil Normalisasi

```
python penyajian.py
Index(['bed', 'bath', 'acre_lot', 'house_size', 'price'], dtype='object')
Data setelah dinormalisasi:
```

	bed	bath	acre_lot	house_size	price
0	0.166667	0.125	0.002223	0.028175	0.004369
1	0.250000	0.125	0.001334	0.058880	0.003085
2	0.083333	0.000	0.002890	0.019475	0.002418
3	0.250000	0.125	0.001779	0.072690	0.006422
4	0.250000	0.250	0.009782	0.109110	0.008167
5	0.166667	0.000	0.004002	0.084830	0.001545
6	0.166667	0.125	0.001334	0.034751	0.002654
7	0.083333	0.000	0.001556	0.036876	0.004112
8	0.333333	0.250	0.165407	0.254945	0.014379
9	0.166667	0.125	0.297243	0.037584	0.003547
10	0.166667	0.125	0.001334	0.034498	0.006679
11	0.166667	0.125	0.001779	0.192119	0.006935
12	0.333333	0.125	0.002223	0.063584	0.003034
13	0.333333	0.500	0.016007	0.117052	0.032294
14	0.166667	0.125	0.001334	0.037281	0.005139
15	0.250000	0.375	0.004446	0.156154	0.011042
16	0.166667	0.125	0.001334	0.057514	0.004369
17	0.166667	0.125	0.085816	0.183975	0.028496
18	0.416667	0.250	0.005113	0.043857	0.006165
19	0.083333	0.000	0.004669	0.013051	0.001545
20	0.416667	0.250	0.001779	0.133391	0.007449
21	0.166667	0.000	0.044020	0.042997	0.008681
22	0.166667	0.125	0.000667	0.038191	0.005267
23	0.166667	0.125	0.005780	0.057514	0.012069
24	0.250000	0.250	0.006225	0.133391	0.020796
25	0.250000	0.125	0.002001	0.047397	0.003753
26	0.250000	0.125	0.000445	0.036269	0.002829
27	0.166667	0.125	0.014896	0.108351	0.023054
28	0.416667	0.250	0.038462	0.079923	0.008681
29	0.333333	0.375	0.002223	0.062573	0.010529

2. Hasil Penyajian Data dan Grafik


```

===== KELOMPOK DATA BED =====
Data dengan Kelompok dan Frekuensinya:
bed Kelompok data bed
0      3      1-3
1      4      4-7
2      2      1-3
3      4      4-7
4      4      4-7
..    ...    ...
95     3      1-3
96     4      4-7
97     3      1-3
98    12     11-13
99     9      8-10

[100 rows x 2 columns]

Frekuensi untuk Setiap Kelompok:
1-3      1163
4-7      845
8-10      96
11-13     15
Name: Kelompok data bed, dtype: int64

Total : 2119

Distribusi Kumulatif Kurang dari:
[('1-3', 1163), ('4-7', 2008), ('8-10', 2104), ('11-13', 2119)]

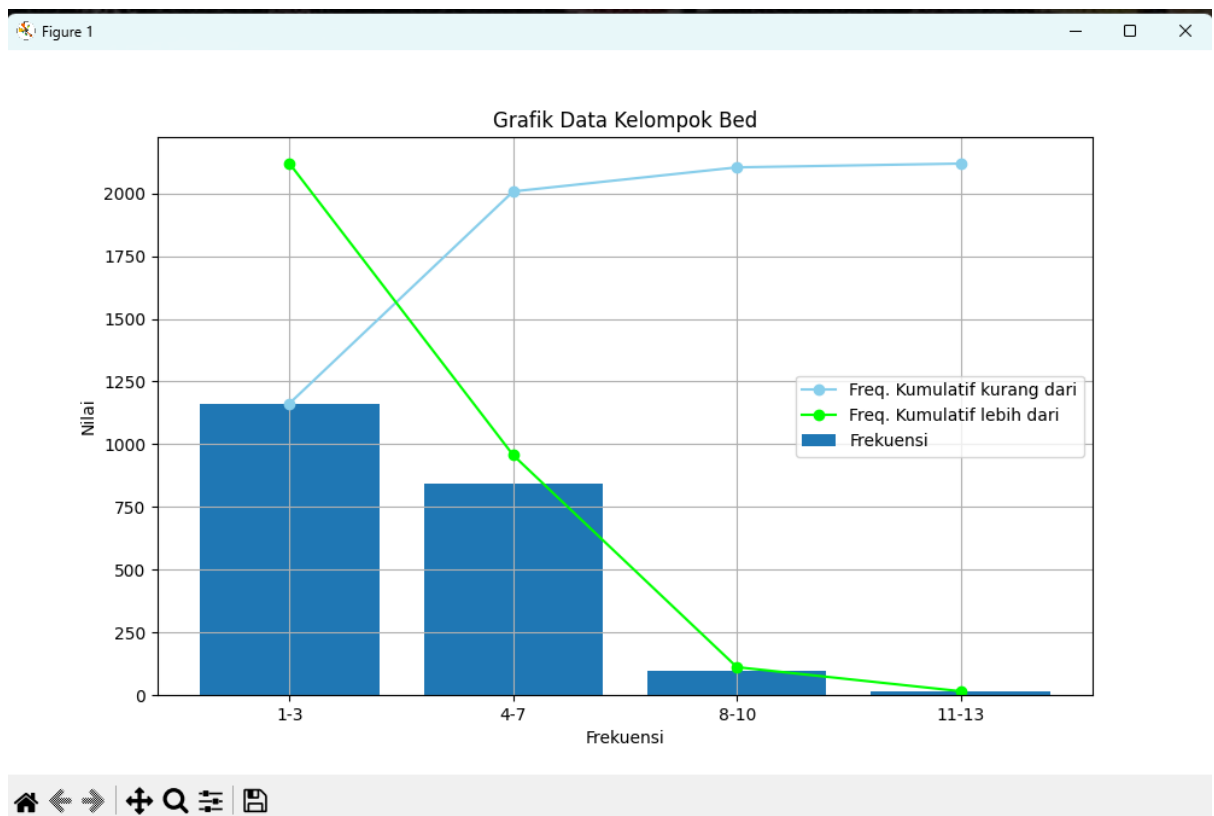
Distribusi Kumulatif Lebih dari:
[('1-3', 2119), ('4-7', 956), ('8-10', 111), ('11-13', 15)]

Distribusi Relatif:
[('1-3', 54.88437942425672), ('4-7', 39.87730061349693), ('8-10', 4.530438886267108), ('11-13', 0.7078810759792354)]

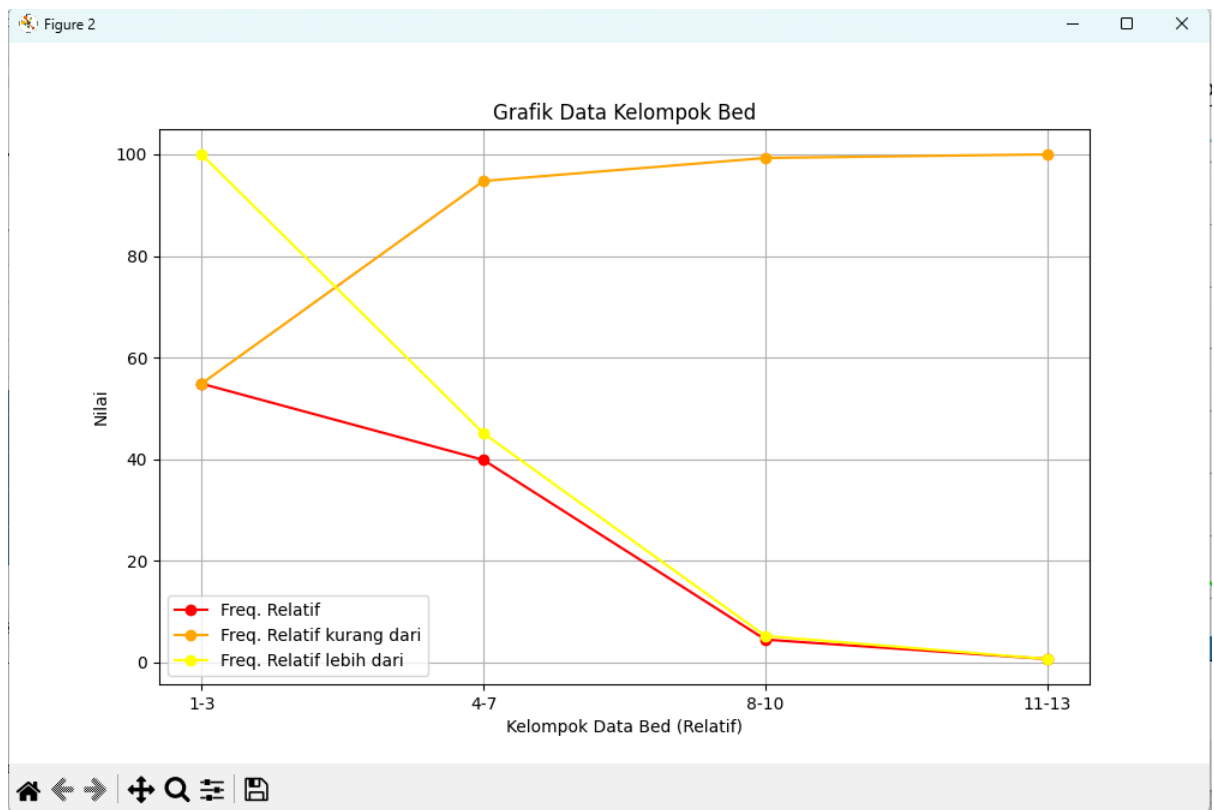
Distribusi Relatif Kumulatif Kurang dari:
[('1-3', 54.88437942425672), ('4-7', 94.76168003775365), ('8-10', 99.29211892402076), ('11-13', 99.99999999999999)]

Distribusi Relatif Kumulatif Lebih dari:
[('1-3', 99.99999999999999), ('4-7', 45.11562057574327), ('8-10', 5.238319962246344), ('11-13', 0.7078810759792354)]

```



(gambar hasil grafik Frekuensi, Frek kumulatif kurang dari dan lebih dari)



(gambar hasil grafik Frekuensi Relatif, Frek Relatif kurang dari dan lebih dari)

D. Analisis Hasil Running Program

Analisis hasil running program penyajian data menunjukkan bahwa output yang dihasilkan dari program ini cukup konsisten dengan hasil yang ada pada excel. Dengan memisah”kan atau mengelompokkan grafik mungkin akan lebih memudahkan pembaca untuk memahami informasi yang disajikan, dengan begitu akan meningkatkan visual data dan mepermudah keterbacaan data yang disajikan, serta dapat dilakukan analisis lebih mendalam dengan lebih mudah.