

# Daftar Isi

1. [Nomor 1: Deskripsi Statistika](#)
2. [Nomor 2: Visualisasi](#)
3. [Nomor 3: Distribusi Normal](#)
4. [Nomor 4: Hipotesis 1 Sampel](#)
5. [Nomor 5: Hipotesis 2 Sampel](#)
6. [Nomor 6: Tes Korelasi](#)

## Nomor 1

### Deskripsi Daerah

```
In [1]: #init
import pandas as pd
import matplotlib.pyplot as plt
#baca file dan memberikan nama
df = pd.read_csv('Gandum.csv', delimiter=',', names = ['id', 'Daerah', 'SumbuUtama', 'SumbuKecil', 'Keunikan', 'AreaBulatan', 'Diameter', 'KadarAir', 'Keliling', 'Buah', 'Warna'])

import numpy as np
import matplotlib.pyplot as plt
import statistics
from scipy import stats
from scipy.stats import norm
from scipy.stats import shapiro
import seaborn as sns
from IPython.display import display, Math, Latex
import math
import random
```

```
In [2]: #NOMOR 1

#print("Daerah, SumbuUtama, SumbuKecil, Keunikan, AreaBulatan, Diameter, KadarAir, Keliling, Buah, Warna")
inp = "Daerah"

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n====",str(inp),"====")
print("Mean dari", str(inp), "adalah", mean)
print("Modus dari", str(inp), "adalah", modus)
print("Standard deviasi dari", str(inp), "adalah", stdev)
print("Variansi dari", str(inp), "adalah", var)
print("Range dari", str(inp), "adalah", range1)
print("Maximum dari", str(inp), "adalah", max1)
print("Minimum dari", str(inp), "adalah", min1)
print("Q1 dari", str(inp), "adalah", q1)
print("Q2 dari", str(inp), "adalah", q2)
```

```

print("Q3 dari", str(inp), "adalah" ,q3)
print("Interkuartil dari", str(inp), "adalah" ,iqr)
print("Skewness dari", str(inp), "adalah" ,skew)
print("Kurtosis dari", str(inp), "adalah" ,kurtosis)

```

=== Daerah ===

```

Mean dari Daerah adalah 4801.246
Modus dari Daerah adalah 4881
Standard deviasi dari Daerah adalah 986.3954914816018
Variansi dari Daerah adalah 972976.0656152307
Range dari Daerah adalah 4931
Maximum dari Daerah adalah 7453
Minimum dari Daerah adalah 2522
Q1 dari Daerah adalah 4042.75
Q2 dari Daerah adalah 4735.0
Q3 dari Daerah adalah 5495.5
Interkuartil dari Daerah adalah 1452.75
Skewness dari Daerah adalah 0.23814408738280812
Kurtosis dari Daerah adalah -0.4346305340273977

```

## Deskripsi SumbuUtama

```

In [3]: inp = "SumbuUtama"

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n===",str(inp),"===")
print("Mean dari", str(inp), "adalah" ,mean)
print("Modus dari", str(inp), "adalah" ,modus)
print("Standard deviasi dari", str(inp), "adalah" ,stdev)
print("Variansi dari", str(inp), "adalah" ,var)
print("Range dari", str(inp), "adalah" ,range1)
print("Maximum dari", str(inp), "adalah" ,max1)
print("Minimum dari", str(inp), "adalah" ,min1)
print("Q1 dari", str(inp), "adalah" ,q1)
print("Q2 dari", str(inp), "adalah" ,q2)
print("Q3 dari", str(inp), "adalah" ,q3)
print("Interkuartil dari", str(inp), "adalah" ,iqr)
print("Skewness dari", str(inp), "adalah" ,skew)
print("Kurtosis dari", str(inp), "adalah" ,kurtosis)

```

=== SumbuUtama ===

```

Mean dari SumbuUtama adalah 116.04517136778003
Modus dari SumbuUtama adalah 99.01478925
Standard deviasi dari SumbuUtama adalah 18.28262595755936
Variansi dari SumbuUtama adalah 334.25441190402336
Range dari SumbuUtama adalah 153.79546865999998
Maximum dari SumbuUtama adalah 227.9285827
Minimum dari SumbuUtama adalah 74.13311404
Q1 dari SumbuUtama adalah 104.11609817499999
Q2 dari SumbuUtama adalah 115.40513994999999
Q3 dari SumbuUtama adalah 129.046792025
Interkuartil dari SumbuUtama adalah 24.93069385000001
Skewness dari SumbuUtama adalah 0.7615287378076631
Kurtosis dari SumbuUtama adalah 4.330533548436968

```

## Deskripsi SumbuKecil

```
In [4]: inp = "SumbuKecil"

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n===",str(inp),"===")
print("Mean dari", str(inp), "adalah" ,mean)
print("Modus dari", str(inp), "adalah" ,modus)
print("Standard deviasi dari", str(inp), "adalah" ,stdev)
print("Variansi dari", str(inp), "adalah" ,var)
print("Range dari", str(inp), "adalah" ,range1)
print("Maximum dari", str(inp), "adalah" ,max1)
print("Minimum dari", str(inp), "adalah" ,min1)
print("Q1 dari", str(inp), "adalah" ,q1)
print("Q2 dari", str(inp), "adalah" ,q2)
print("Q3 dari", str(inp), "adalah" ,q3)
print("Interkuartil dari", str(inp), "adalah" ,iqr)
print("Skewness dari", str(inp), "adalah" ,skew)
print("Kurtosis dari", str(inp), "adalah" ,kurtosis)
```

```
=== SumbuKecil ===
Mean dari SumbuKecil adalah 53.71524598896002
Modus dari SumbuKecil adalah 53.17603697
Standard deviasi dari SumbuKecil adalah 4.071074752475035
Variansi dari SumbuKecil adalah 16.573649640239662
Range dari SumbuKecil adalah 29.071182429999993
Maximum dari SumbuKecil adalah 68.97769987
Minimum dari SumbuKecil adalah 39.90651744
Q1 dari SumbuKecil adalah 51.1935763325
Q2 dari SumbuKecil adalah 53.731198774999996
Q3 dari SumbuKecil adalah 56.3251579825
Interkuartil dari SumbuKecil adalah 5.131581650000001
Skewness dari SumbuKecil adalah -0.010828051555611359
Kurtosis dari SumbuKecil adalah 0.47556845043813656
```

## Deskripsi Keunikan

```
In [5]: inp = "Keunikan"

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n===",str(inp),"===")
print("Mean dari", str(inp), "adalah" ,mean)
```

```

print("Modus dari", str(inp), "adalah" ,modus)
print("Standard deviasi dari", str(inp), "adalah" ,stdev)
print("Variansi dari", str(inp), "adalah" ,var)
print("Range dari", str(inp), "adalah" ,range1)
print("Maximum dari", str(inp), "adalah" ,max1)
print("Minimum dari", str(inp), "adalah" ,min1)
print("Q1 dari", str(inp), "adalah" ,q1)
print("Q2 dari", str(inp), "adalah" ,q2)
print("Q3 dari", str(inp), "adalah" ,q3)
print("Interkuartil dari", str(inp), "adalah" ,iqr)
print("Skewness dari", str(inp), "adalah" ,skew)
print("Kurtosis dari", str(inp), "adalah" ,kurtosis)

```

```

=== Keunikan ===
Mean dari Keunikan adalah 0.878763914340001
Modus dari Keunikan adalah 0.90928407
Standard deviasi dari Keunikan adalah 0.03658590550477702
Variansi dari Keunikan adalah 0.0013385284816044736
Range dari Keunikan adalah 0.19408517999999997
Maximum dari Keunikan adalah 0.91400140599999999
Minimum dari Keunikan adalah 0.71991622599999999
Q1 dari Keunikan adalah 0.8636757527500001
Q2 dari Keunikan adalah 0.8900454185
Q3 dari Keunikan adalah 0.907577917
Interkuartil dari Keunikan adalah 0.04390216424999993
Skewness dari Keunikan adalah -1.6234718222806501
Kurtosis dari Keunikan adalah 2.917255925694389

```

## Deskripsi AreaBulatan

```

In [6]: inp = "AreaBulatan"

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n===",str(inp),"===")
print("Mean dari", str(inp), "adalah" ,mean)
print("Modus dari", str(inp), "adalah" ,modus)
print("Standard deviasi dari", str(inp), "adalah" ,stdev)
print("Variansi dari", str(inp), "adalah" ,var)
print("Range dari", str(inp), "adalah" ,range1)
print("Maximum dari", str(inp), "adalah" ,max1)
print("Minimum dari", str(inp), "adalah" ,min1)
print("Q1 dari", str(inp), "adalah" ,q1)
print("Q2 dari", str(inp), "adalah" ,q2)
print("Q3 dari", str(inp), "adalah" ,q3)
print("Interkuartil dari", str(inp), "adalah" ,iqr)
print("Skewness dari", str(inp), "adalah" ,skew)
print("Kurtosis dari", str(inp), "adalah" ,kurtosis)

```

```

=== AreaBulatan ===
Mean dari AreaBulatan adalah 4937.048
Modus dari AreaBulatan adalah 3802
Standard deviasi dari AreaBulatan adalah 1011.6962549701573
Variansi dari AreaBulatan adalah 1023529.3123206415
Range dari AreaBulatan adalah 5141
Maximum dari AreaBulatan adalah 7720
Minimum dari AreaBulatan adalah 2579

```

Q1 dari AreaBulatan adalah 4170.25  
Q2 dari AreaBulatan adalah 4857.0  
Q3 dari AreaBulatan adalah 5654.25  
Interkuartil dari AreaBulatan adalah 1484.0  
Skewness dari AreaBulatan adalah 0.2575600053152032  
Kurtosis dari AreaBulatan adalah -0.40968492037366033

## Deskripsi Diameter

```
In [7]: inp = "Diameter"

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n===",str(inp),"===")
print("Mean dari", str(inp), "adalah", mean)
print("Modus dari", str(inp), "adalah", modus)
print("Standard deviasi dari", str(inp), "adalah", stdev)
print("Variansi dari", str(inp), "adalah", var)
print("Range dari", str(inp), "adalah", range1)
print("Maximum dari", str(inp), "adalah", max1)
print("Minimum dari", str(inp), "adalah", min1)
print("Q1 dari", str(inp), "adalah", q1)
print("Q2 dari", str(inp), "adalah", q2)
print("Q3 dari", str(inp), "adalah", q3)
print("Interkuartil dari", str(inp), "adalah", iqr)
print("Skewness dari", str(inp), "adalah", skew)
print("Kurtosis dari", str(inp), "adalah", kurtosis)

=== Diameter ===
Mean dari Diameter adalah 77.77115780832005
Modus dari Diameter adalah 78.83325579
Standard deviasi dari Diameter adalah 8.056867291849715
Variansi dari Diameter adalah 64.91311055847777
Range dari Diameter adalah 40.747172240000005
Maximum dari Diameter adalah 97.41383027
Minimum dari Diameter adalah 56.66665803
Q1 dari Diameter adalah 71.74530754749999
Q2 dari Diameter adalah 77.64527658
Q3 dari Diameter adalah 83.6485975675
Interkuartil dari Diameter adalah 11.903290020000014
Skewness dari Diameter adalah 0.002724966865193717
Kurtosis dari Diameter adalah -0.4664545130312141
```

## Deskripsi KadarAir

```
In [8]: inp = "KadarAir"

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
```

```

q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n===",str(inp),"===")
print("Mean dari", str(inp), "adalah", mean)
print("Modus dari", str(inp), "adalah", modus)
print("Standard deviasi dari", str(inp), "adalah", stdev)
print("Variansi dari", str(inp), "adalah", var)
print("Range dari", str(inp), "adalah", range1)
print("Maximum dari", str(inp), "adalah", max1)
print("Minimum dari", str(inp), "adalah", min1)
print("Q1 dari", str(inp), "adalah", q1)
print("Q2 dari", str(inp), "adalah", q2)
print("Q3 dari", str(inp), "adalah", q3)
print("Interkuartil dari", str(inp), "adalah", iqr)
print("Skewness dari", str(inp), "adalah", skew)
print("Kurtosis dari", str(inp), "adalah", kurtosis)

```

```

=== KadarAir ===
Mean dari KadarAir adalah 0.6483716718980002
Modus dari KadarAir adalah 0.824404762
Standard deviasi dari KadarAir adalah 0.0943670980937953
Variansi dari KadarAir adalah 0.008905149202643984
Range dari KadarAir adalah 0.46897193099999995
Maximum dari KadarAir adalah 0.878899083
Minimum dari KadarAir adalah 0.409927152
Q1 dari KadarAir adalah 0.57263245725
Q2 dari KadarAir adalah 0.626116699
Q3 dari KadarAir adalah 0.7266333445000001
Interkuartil dari KadarAir adalah 0.15400088725000005
Skewness dari KadarAir adalah 0.49366131797330265
Kurtosis dari KadarAir adalah -0.7403261705867821

```

## Deskripsi Keliling

```

In [9]: inp = "Keliling"

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n===",str(inp),"===")
print("Mean dari", str(inp), "adalah", mean)
print("Modus dari", str(inp), "adalah", modus)
print("Standard deviasi dari", str(inp), "adalah", stdev)
print("Variansi dari", str(inp), "adalah", var)
print("Range dari", str(inp), "adalah", range1)
print("Maximum dari", str(inp), "adalah", max1)
print("Minimum dari", str(inp), "adalah", min1)
print("Q1 dari", str(inp), "adalah", q1)
print("Q2 dari", str(inp), "adalah", q2)
print("Q3 dari", str(inp), "adalah", q3)
print("Interkuartil dari", str(inp), "adalah", iqr)

```

```
print("Skewness dari", str(inp), "adalah" ,skew)
print("Kurtosis dari", str(inp), "adalah" ,kurtosis)
```

```
=== Keliling ===
Mean dari Keliling adalah 281.479722
Modus dari Keliling adalah 202.456
Standard deviasi dari Keliling adalah 37.335401715143995
Variansi dari Keliling adalah 1393.9322212311772
Range dari Keliling adalah 291.822
Maximum dari Keliling adalah 488.837
Minimum dari Keliling adalah 197.015
Q1 dari Keliling adalah 255.88299999999998
Q2 dari Keliling adalah 280.04549999999995
Q3 dari Keliling adalah 306.0625
Interkuartil dari Keliling adalah 50.17950000000002
Skewness dari Keliling adalah 0.7336269072005543
Kurtosis dari Keliling adalah 2.272684731245571
```

## Deskripsi Bulatan

```
In [10]: inp = "Bulatan"

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n===",str(inp),"===")
print("Mean dari", str(inp), "adalah" ,mean)
print("Modus dari", str(inp), "adalah" ,modus)
print("Standard deviasi dari", str(inp), "adalah" ,stdev)
print("Variansi dari", str(inp), "adalah" ,var)
print("Range dari", str(inp), "adalah" ,range1)
print("Maximum dari", str(inp), "adalah" ,max1)
print("Minimum dari", str(inp), "adalah" ,min1)
print("Q1 dari", str(inp), "adalah" ,q1)
print("Q2 dari", str(inp), "adalah" ,q2)
print("Q3 dari", str(inp), "adalah" ,q3)
print("Interkuartil dari", str(inp), "adalah" ,iqr)
print("Skewness dari", str(inp), "adalah" ,skew)
print("Kurtosis dari", str(inp), "adalah" ,kurtosis)
```

```
=== Bulatan ===
Mean dari Bulatan adalah 0.7617374363080005
Modus dari Bulatan adalah 0.7556490159999999
Standard deviasi dari Bulatan adalah 0.0617024607867326
Variansi dari Bulatan adalah 0.0038071936671382743
Range dari Bulatan adalah 0.7301581349999999
Maximum dari Bulatan adalah 0.9047483129999999
Minimum dari Bulatan adalah 0.174590178
Q1 dari Bulatan adalah 0.731990728
Q2 dari Bulatan adalah 0.7612884395
Q3 dari Bulatan adalah 0.79636096975
Interkuartil dari Bulatan adalah 0.06437024175000006
Skewness dari Bulatan adalah -3.5992367663616407
Kurtosis dari Bulatan adalah 29.97509590463005
```

## Deskripsi Ransum

```
In [11]: inp = "Ransum"
```

```

mean = df[inp].mean()
median = df[inp].median()
modus = df[inp].value_counts().idxmax()
stdev = df[inp].std()
var = df[inp].var()
min1 = df[inp].min()
max1 = df[inp].max()
range1 = max1 - min1
q1 = df[inp].quantile(0.25)
q2 = df[inp].quantile(0.5)
q3 = df[inp].quantile(0.75)
iqr = q3 - q1
skew = df[inp].skew()
kurtosis = df[inp].kurtosis()

#print
print("\n===",str(inp),"===")
print("Mean dari", str(inp), "adalah" ,mean)
print("Modus dari", str(inp), "adalah" ,modus)
print("Standard deviasi dari", str(inp), "adalah" ,stdev)
print("Variansi dari", str(inp), "adalah" ,var)
print("Range dari", str(inp), "adalah" ,range1)
print("Maximum dari", str(inp), "adalah" ,max1)
print("Minimum dari", str(inp), "adalah" ,min1)
print("Q1 dari", str(inp), "adalah" ,q1)
print("Q2 dari", str(inp), "adalah" ,q2)
print("Q3 dari", str(inp), "adalah" ,q3)
print("Interkuartil dari", str(inp), "adalah" ,iqr)
print("Skewness dari", str(inp), "adalah" ,skew)
print("Kurtosis dari", str(inp), "adalah" ,kurtosis)

```

```

=== Ransum ===
Mean dari Ransum adalah 2.150915331084002
Modus dari Ransum adalah 1.901437269
Standard deviasi dari Ransum adalah 0.24976688927065827
Variansi dari Ransum adalah 0.06238349897594127
Range dari Ransum adalah 1.0240129659999997
Maximum dari Ransum adalah 2.4648085809999998
Minimum dari Ransum adalah 1.440795615
Q1 dari Ransum adalah 1.98393879075
Q2 dari Ransum adalah 2.1935990365
Q3 dari Ransum adalah 2.38161221825
Interkuartil dari Ransum adalah 0.3976734275
Skewness dari Ransum adalah -0.6581880925333653
Kurtosis dari Ransum adalah -0.4286557930626156

```

## Nomor 2

### Visualisasi Daerah

```

In [12]: #NOMOR 2

#inp = input("Masukkan nama data yang dicari : ")
inp = "Daerah"

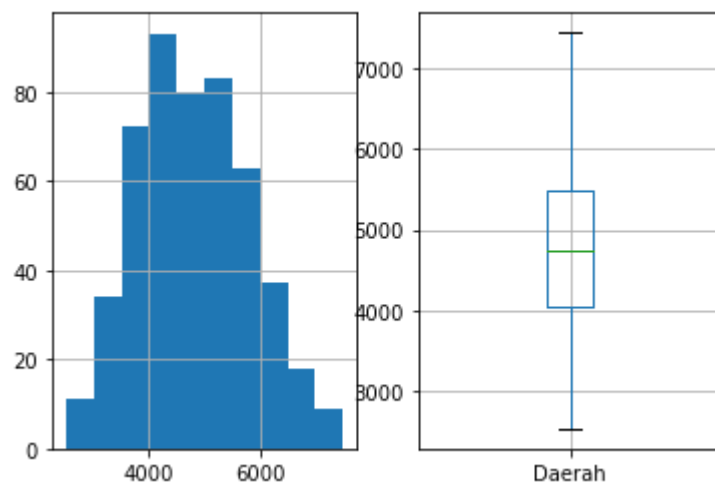
fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)

```

Out[12]: <AxesSubplot:>





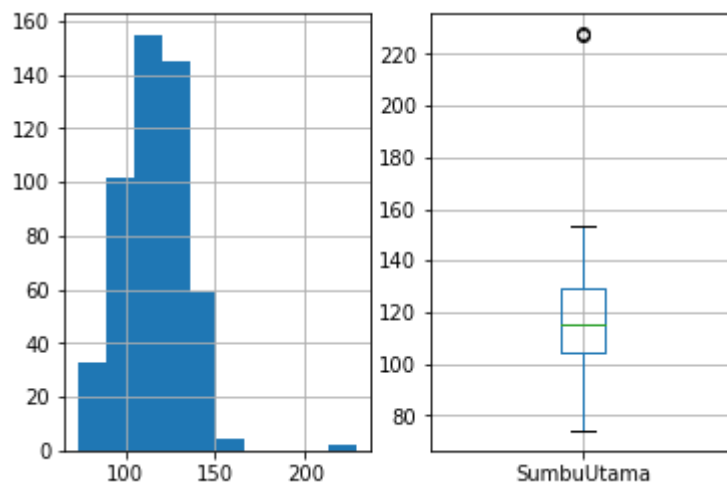
## Visualisasi SumbuUtama

```
In [13]: inp = "SumbuUtama"

fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)
```

Out[13]: <AxesSubplot:>



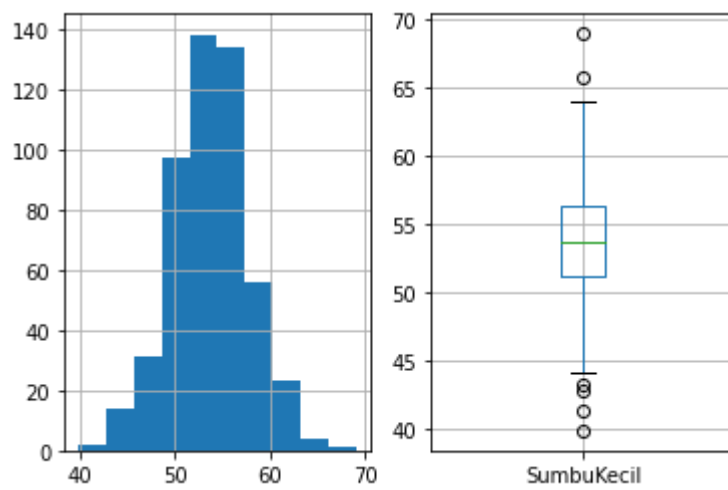
## Visualisasi SumbuKecil

```
In [14]: inp = "SumbuKecil"

fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)
```

Out[14]: <AxesSubplot:>



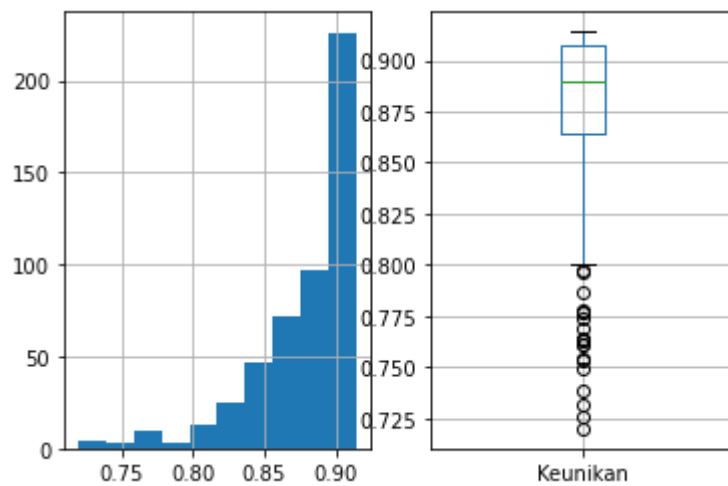
## Visualisasi Keunikan

```
In [15]: inp = "Keunikan"

fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)
```

Out[15]: <AxesSubplot:>



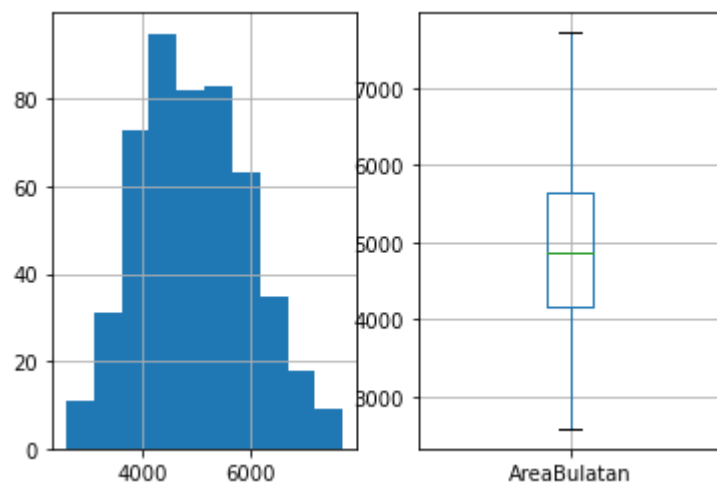
## Visualisasi AreaBulatan

```
In [16]: inp = "AreaBulatan"

fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)
```

Out[16]: <AxesSubplot:>



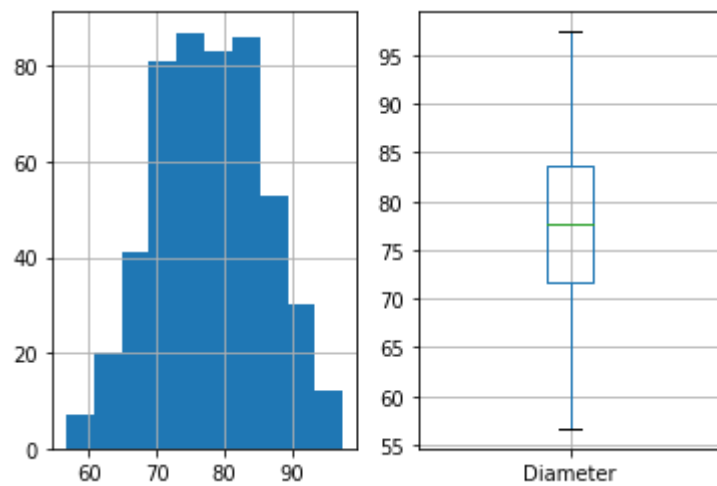
## Visualisasi Diameter

```
In [17]: inp = "Diameter"

fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)
```

Out[17]: <AxesSubplot:>



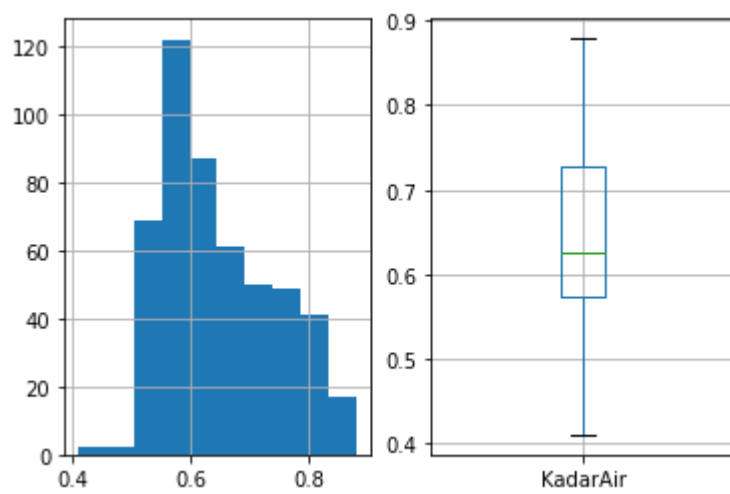
## Visualisasi KadarAir

```
In [18]: inp = "KadarAir"

fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)
```

Out[18]: <AxesSubplot:>



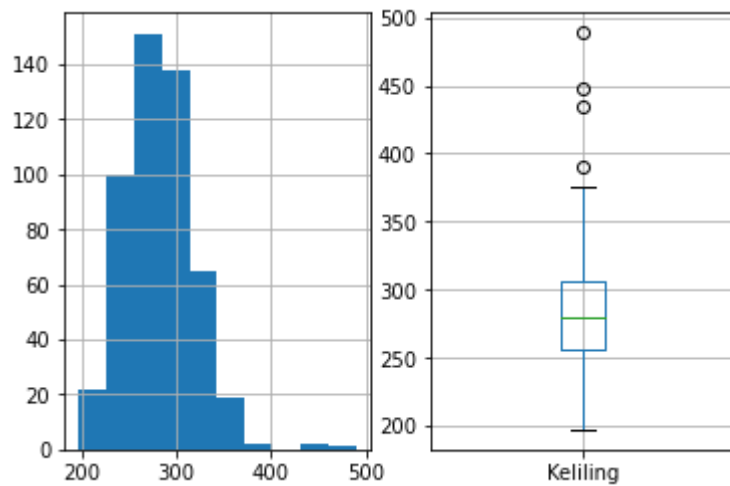
## Visualisasi Keliling

```
In [19]: inp = "Keliling"

fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)
```

Out[19]: <AxesSubplot:>



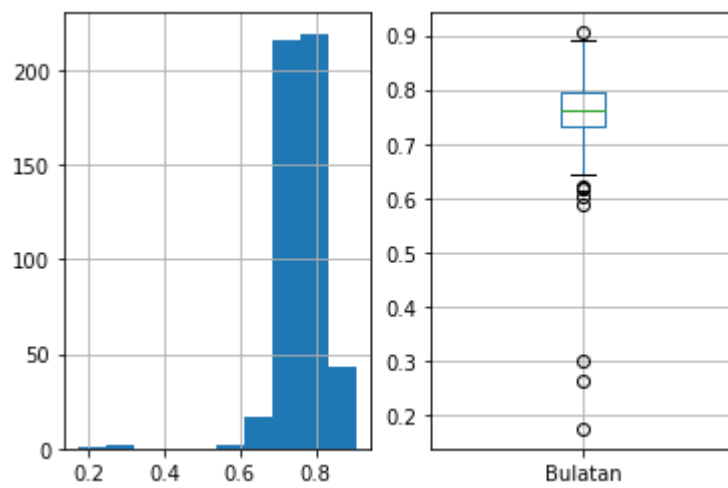
## Visualisasi Bulatan

```
In [20]: inp = "Bulatan"

fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)
```

Out[20]: <AxesSubplot:>



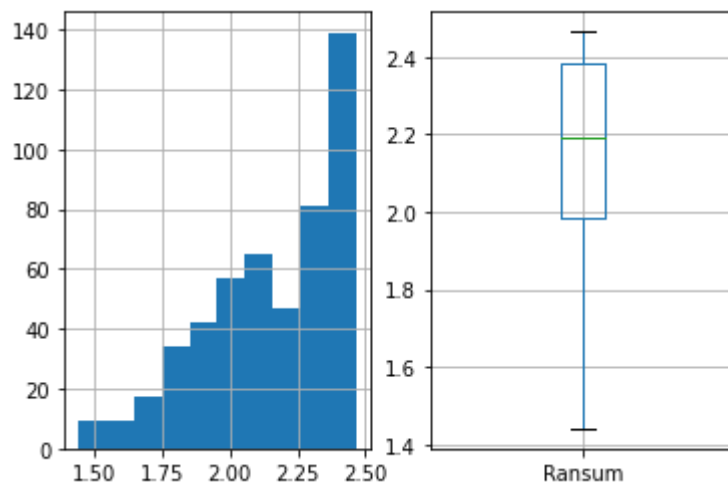
## Visualisasi Ransum

```
In [21]: inp = "Ransum"

fig, axs = plt.subplots(1,2)

df[inp].hist(ax=axs[0])
df.boxplot(inp)
```

Out[21]: <AxesSubplot:>



## Nomor 3

### Tes Daerah

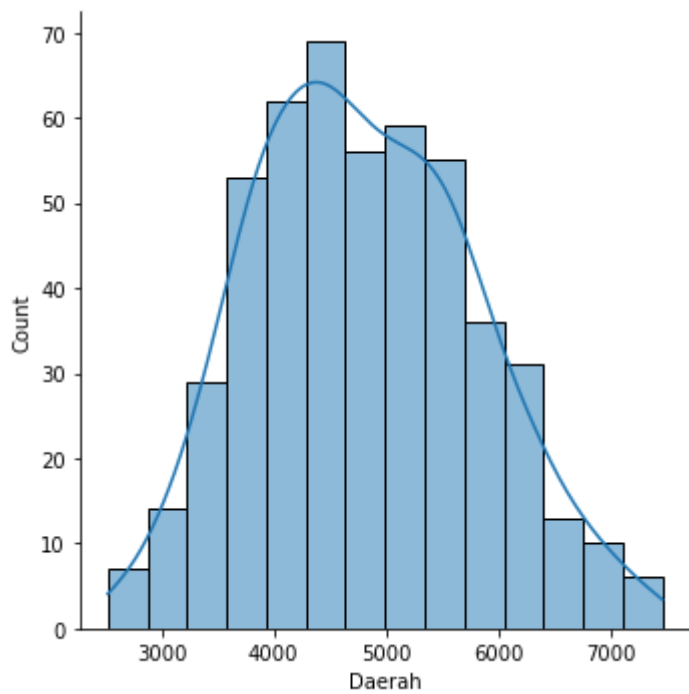
```
In [22]: #NOMOR 3

#inpnorm = input("Masukkan data yang dicari: ")
inpnorm = "Daerah"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')
```

Statistics=0.991, p=0.003  
Tidak Distribusi Normal



## Tes SumbuUtama

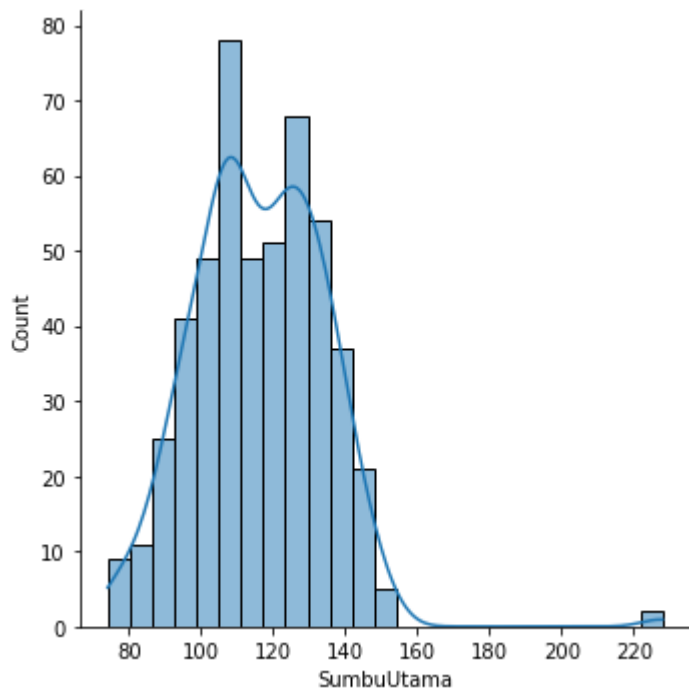
In [23]:

```
inpnorm = "SumbuUtama"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')
```

Statistics=0.951, p=0.000  
Tidak Distribusi Normal



## Tes SumbuKecil

In [24]:

```
inpnorm = "SumbuKecil"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)
```

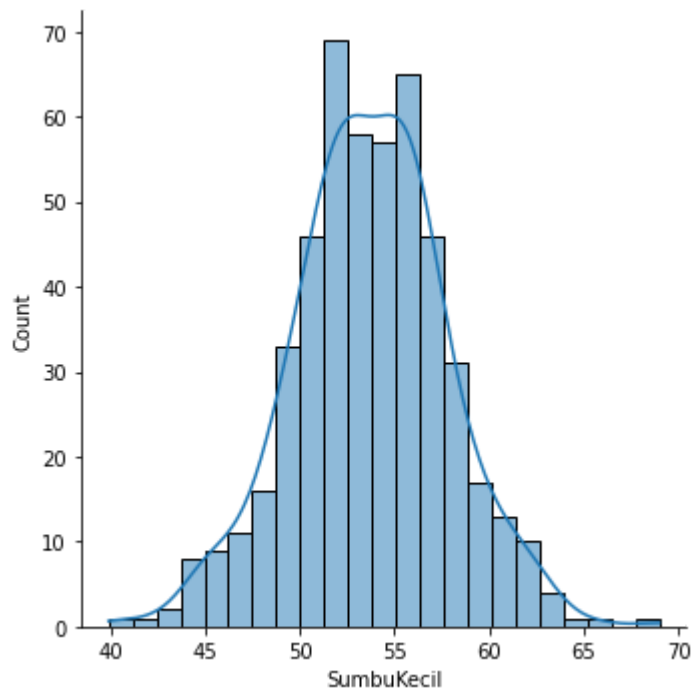
```

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')

```

Statistics=0.997, p=0.423  
Distribusi Normal



## Tes Keunikan

```

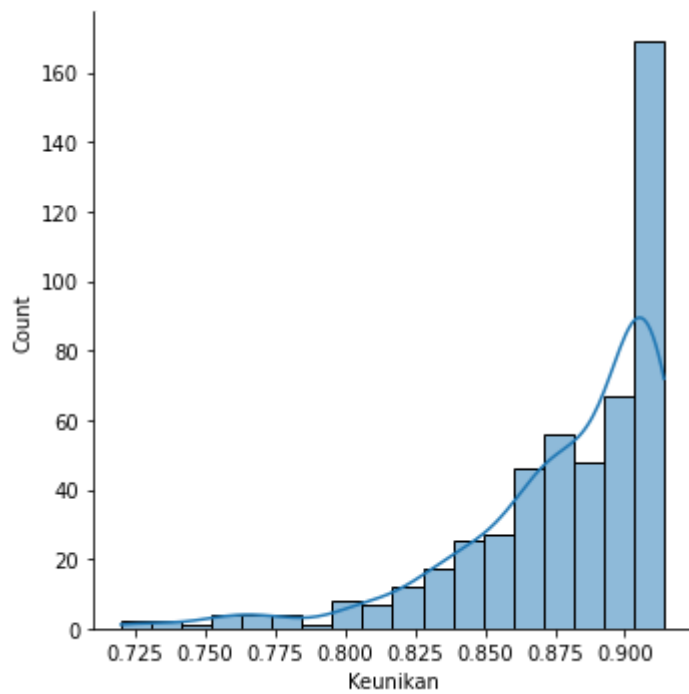
In [25]: inpnorm = "Keunikan"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')

```

Statistics=0.831, p=0.000  
Tidak Distribusi Normal



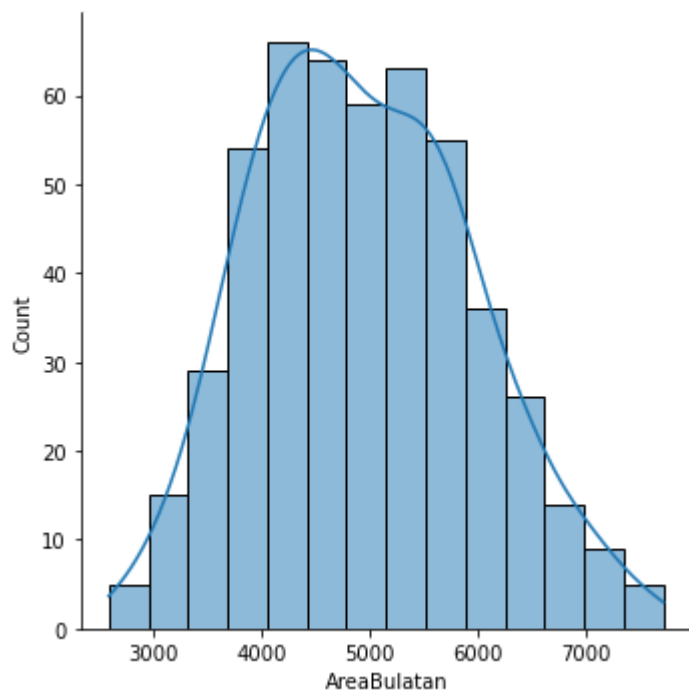
## Tes AreaBulatan

```
In [26]: inpnorm = "AreaBulatan"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')
```

Statistics=0.990, p=0.002  
Tidak Distribusi Normal



## Tes Diameter

```
In [27]: inpnorm = "Diameter"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)
```



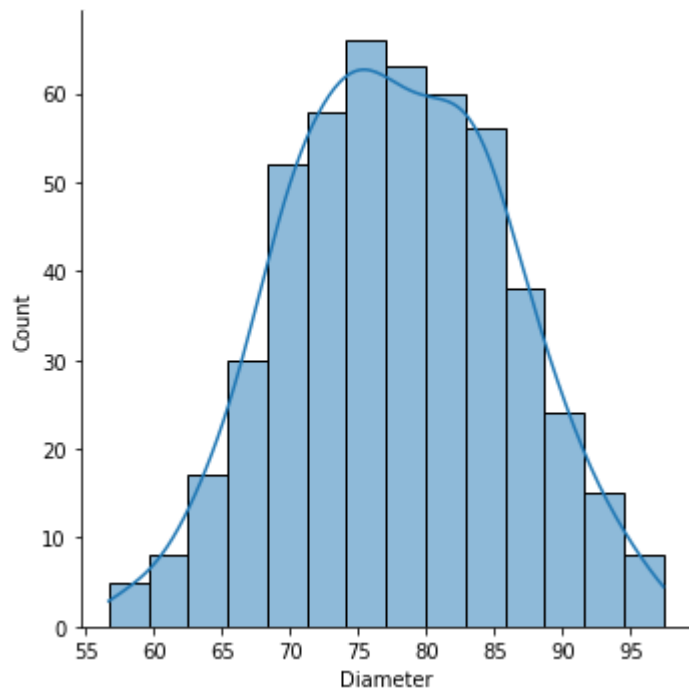
```

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')

```

Statistics=0.995, p=0.118  
Distribusi Normal



## Tes KadarAir

```

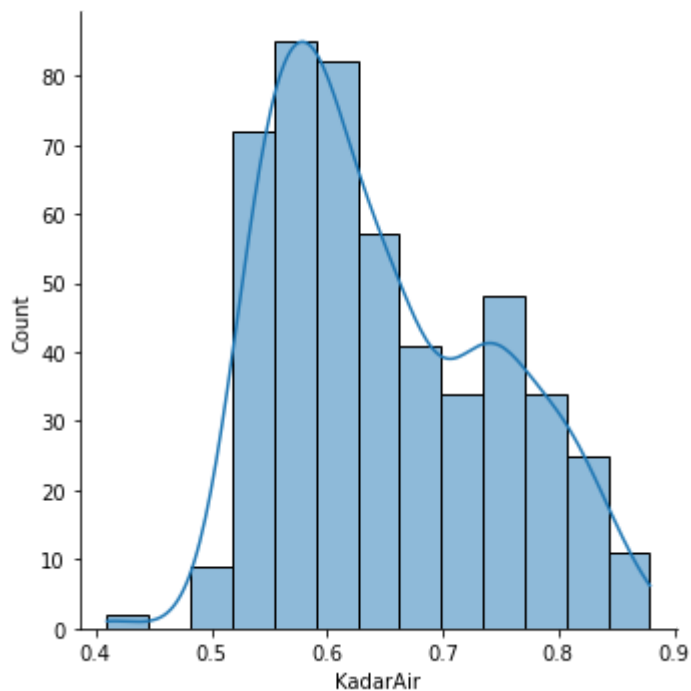
In [28]: inpnorm = "KadarAir"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')

```

Statistics=0.947, p=0.000  
Tidak Distribusi Normal



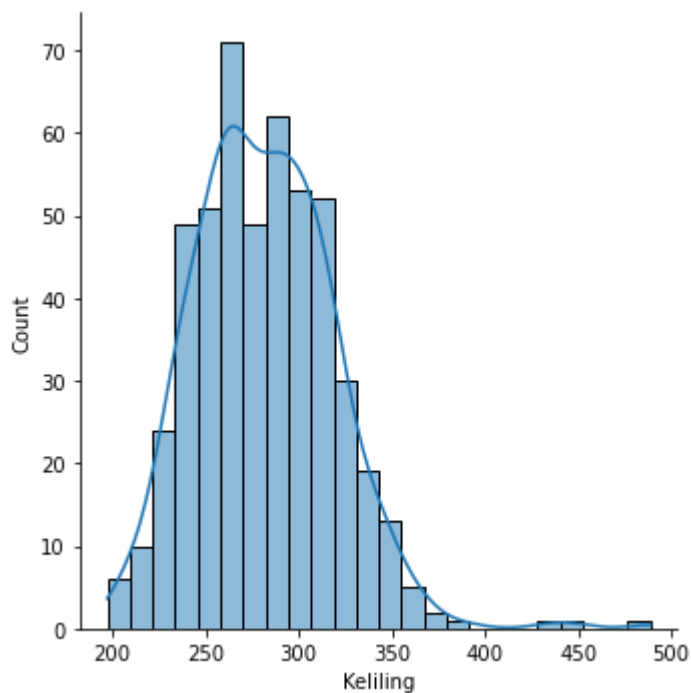
## Tes Keliling

```
In [29]: inpnorm = "Keliling"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')
```

Statistics=0.969, p=0.000  
Tidak Distribusi Normal



## Tes Bulatan

```
In [30]: inpnorm = "Bulatan"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)
```

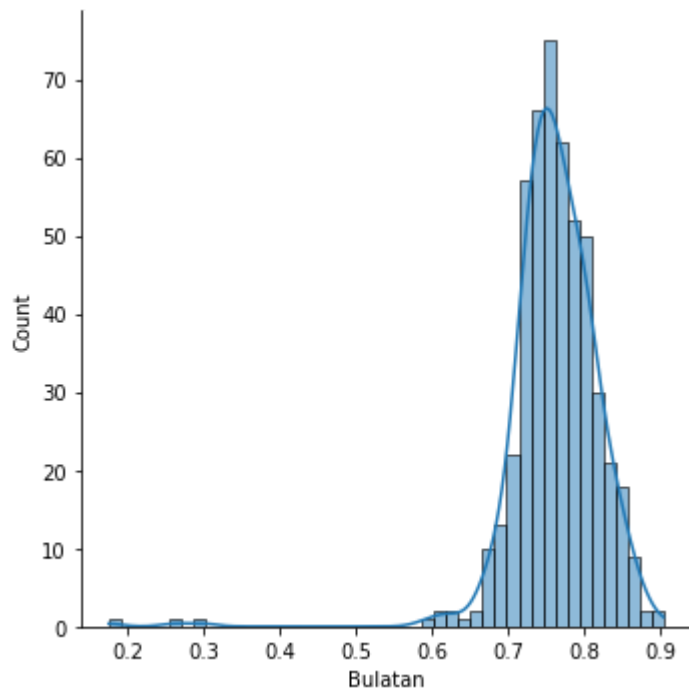
```

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')

```

Statistics=0.770, p=0.000  
Tidak Distribusi Normal



## Tes Ransum

```

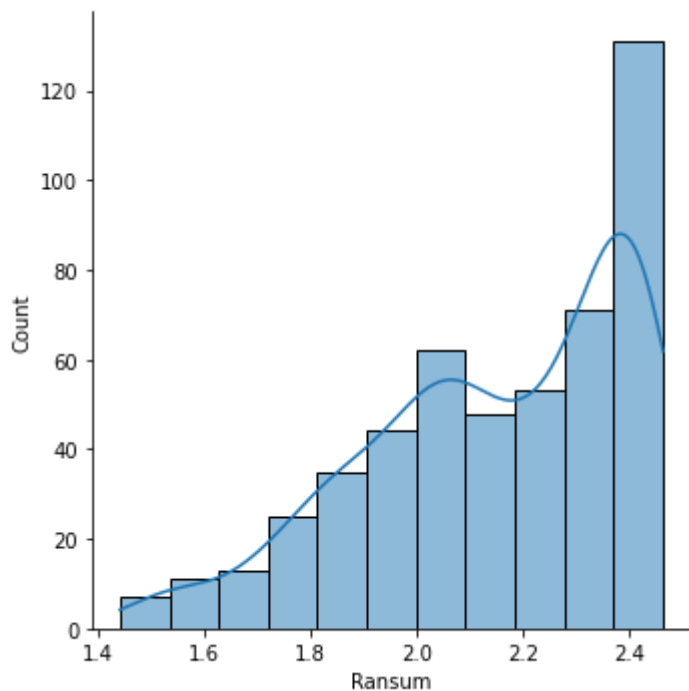
In [31]: inpnorm = "Ransum"
x_axis = df[inpnorm] # kolom yang dituju
ax = sns.displot(x_axis, kde=True)

stat, p = shapiro(x_axis)
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Distribusi Normal')
else:
    print('Tidak Distribusi Normal')

```

Statistics=0.927, p=0.000  
Tidak Distribusi Normal



## Nomor 4

```
In [32]: #NOMOR 4a: nilai rata-rata Daerah di atas 4700

#Sampel yang diambil adalah 100 record random
ukuransampel = 100
alpha = 0.05
skip = sorted(random.sample(range(500),500-ukuransampel))
sampel = pd.read_csv('Gandum.csv', skiprows=skip, delimiter=',', names = ['id', 'Daerah', 'Sumt

print("Langkah 1: Menentukan Hipotesis Nol")
display(Math(r'H_{0}: \mu = 4700'))

print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: \mu > 4700'))

print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
display(Math(f'\alpha = {{{alpha}}}')

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
z = round(stats.norm.ppf(1-alpha),3)
display(Math(f'z > {{{z}}}')
display(Math(r'z = \frac{\overline{x}-\mu_{0}}{\frac{\sigma}{\sqrt{n}}}')

print("\nLangkah 5: Nilai Uji Statistik")
xbar = sampel["Daerah"].mean()
mean = df["Daerah"].mean()
variansi = df["Daerah"].std()
display(Math(f'\overline{x} = {{{xbar}}}')
display(Math(f'\mu_{0} = {{{mean}}}')
display(Math(f'\sigma = {{{variansi}}}')
display(Math(f'n = {{{ukuransampel}}}')
hasil = (xbar-mean)/(variansi/math.sqrt(ukuransampel))
display(Math(f'z = \frac{{{xbar}} - {{{mean}}}}{{\frac{{{variansi}}}{\sqrt{{{ukur

#Hitung p-value
pvalue = stats.norm.sf(abs(hasil))
display(Math(f'P = {{{pvalue}}}')

print("\nLangkah 6: Keputusan")
if(pvalue<alpha):
    display(Math(r'\therefore H_{0} \text{ ditolak karena } P < \alpha .'))
else:
    display(Math(r'\therefore H_{0} \text{ tidak ditolak karena } P > \alpha .'))
```

```
#boxplot
sampel.boxplot("Daerah").set_title("Sampel")
plt.show()
df.boxplot("Daerah").set_title("Populasi")
plt.show()
```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : \mu = 4700$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : \mu > 4700$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

$$z > 1.645$$

$$z = \frac{\bar{x} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$$

Langkah 5: Nilai Uji Statistik

$$\bar{x} = 4969.59$$

$$\mu_0 = 4801.246$$

$$\sigma = 986.3954914816018$$

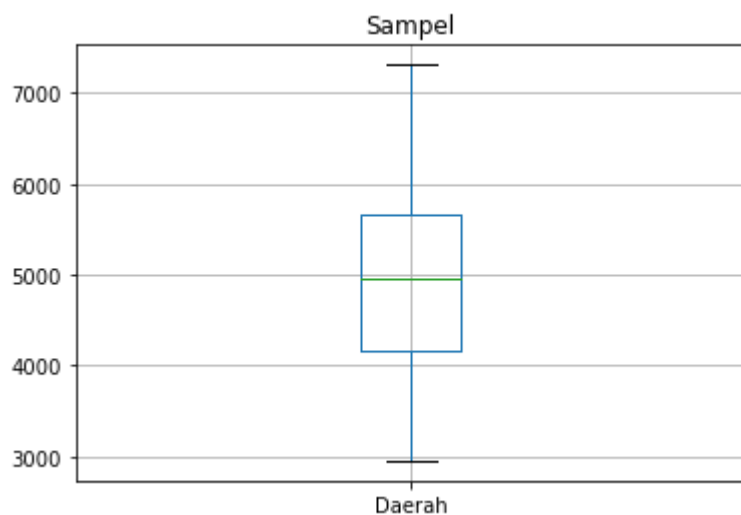
$$n = 100$$

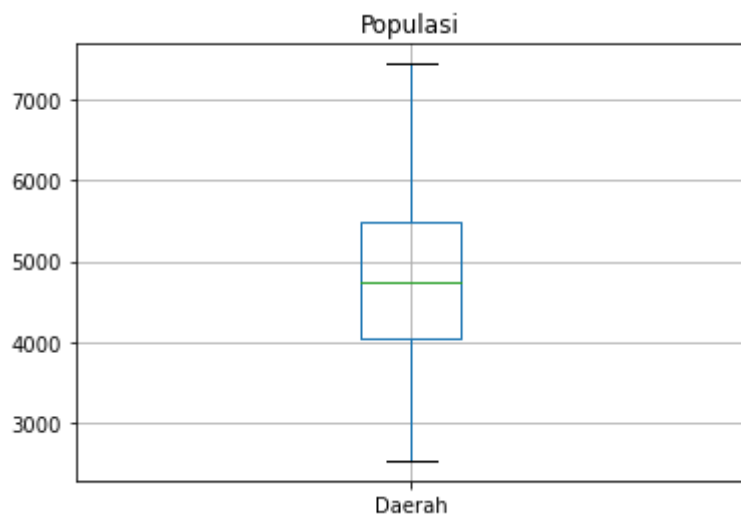
$$z = \frac{4969.59 - 4801.246}{\frac{986.3954914816018}{\sqrt{100}}} = 1.7066582466546079$$

$$P = 0.04394279605366433$$

Langkah 6: Keputusan

$\therefore H_0$  ditolak karena  $P < \alpha$ .





In [33]:

```
#NOMOR 4b: nilai rata-rata SumbuUtama /= 116
#Sampel yang diambil adalah 100 record random

mean = df["SumbuUtama"].mean()
print("Langkah 1: Menentukan Hipotesis Nol")
display(Math(r'H_{0}: \mu = 116'))
print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: \mu \neq 116'))
print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
display(Math(f'\alpha = {{{alpha}}}'))

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
alpha2 = alpha/2
z = round(stats.norm.ppf(1-alpha2),3)
display(Math(f'z > {{{z}}} \text{ atau } z < {{{-z}}}'))
display(Math(r'z = \frac{\overline{x} - \mu_{0}}{\frac{\sigma}{\sqrt{n}}}))

print("\nLangkah 5: Nilai Uji Statistik")
xbar = sampel["SumbuUtama"].mean()
mean = df["SumbuUtama"].mean()
variansi = df["SumbuUtama"].std()
display(Math(f'\overline{x} = {{{xbar}}}'))
display(Math(f'\mu_{0} = {{{mean}}}'))
display(Math(f'\sigma = {{{variansi}}}'))
display(Math(f'n = {{{ukuransampel}}}'))
hasil = (xbar-mean)/(variansi/math.sqrt(ukuransampel))
display(Math(f'z = \frac{{{xbar}} - {{{mean}}}{\frac{{{variansi}}}{\sqrt{{{ukur}}}}}))

#Hitung p-value
pvalue = 2*stats.norm.sf(abs(hasil))
display(Math(f'P = {{{pvalue}}}'))

print("\nLangkah 6: Keputusan")
if(pvalue<alpha):
    display(Math(r'\therefore H_{0} \text{ ditolak karena } P < \alpha .'))
else:
    display(Math(r'\therefore H_{0} \text{ tidak ditolak karena } P > \alpha .'))

#boxplot
sampel.boxplot("SumbuUtama").set_title("Sampel")
plt.show()
df.boxplot("SumbuUtama").set_title("Populasi")
plt.show()
```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : \mu = 116$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : \mu \neq 116$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

$$z > 1.96 \text{ atau } z < -1.96$$

$$z = \frac{\bar{x} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$$

Langkah 5: Nilai Uji Statistik

$$\bar{x} = 118.31762579160002$$

$$\mu_0 = 116.04517136778003$$

$$\sigma = 18.28262595755936$$

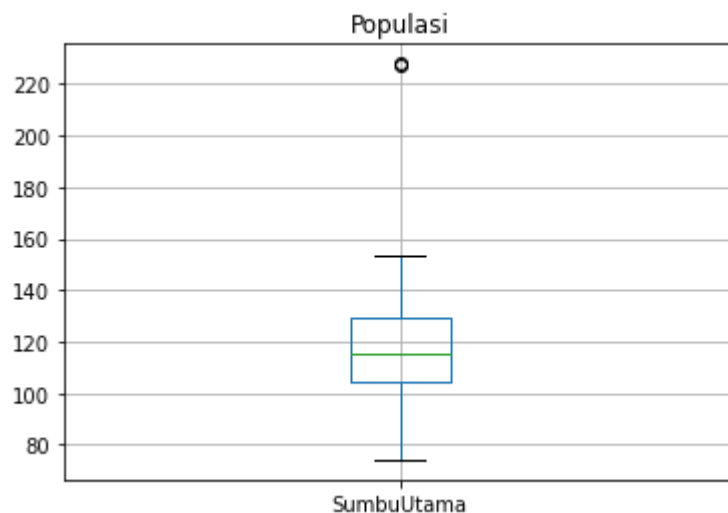
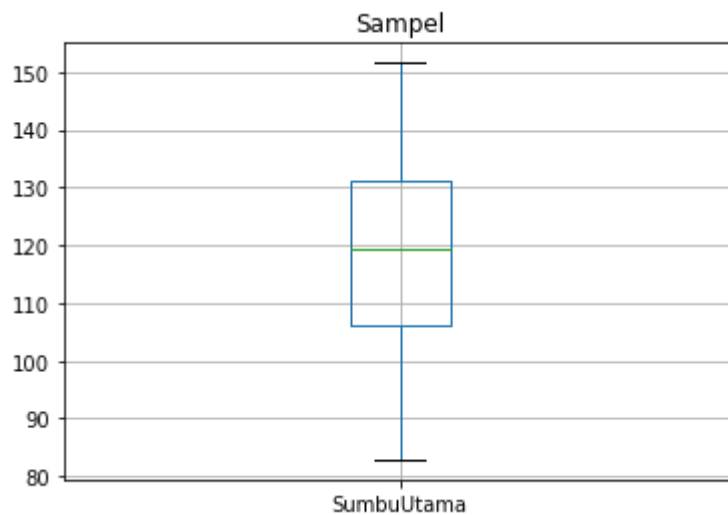
$$n = 100$$

$$z = \frac{118.31762579160002 - 116.04517136778003}{\frac{18.28262595755936}{\sqrt{100}}} = 1.2429584399392017$$

$$P = 0.21388314872468717$$

Langkah 6: Keputusan

$\therefore H_0$  tidak ditolak karena  $P > \alpha$ .



```
In [34]: #NOMOR 4c: rata-rata 20 baris pertama SumbuKecil bukan 50
first20 = df.head(20)
skipsampel = sorted(random.sample(range(20),20-5))
sampel2 = pd.read_csv('Gandum.csv', skiprows=skipsampel, delimiter=',', names = ['id', 'Daerah'])
sampel2 = sampel2.head(5)

print("Langkah 1: Menentukan Hipotesis Nol")
display(Math(r'H_{0}: \mu = 50'))
print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: \mu \neq 50'))
print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
```

```

display(Math(f'\alpha = {{{alpha}}}')')

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
alpha2 = alpha/2
z = round(stats.norm.ppf(1-alpha2),3)
display(Math(f'z > {{{z}}} \text{{ atau }} z < {{{-z}}}')')
display(Math(r'z = \frac{\overline{x} - \mu_{0}}{\frac{\sigma}{\sqrt{n}}}')')

print("\nLangkah 5: Nilai Uji Statistik")
xbar = sampel2["SumbuKecil"].mean()
mean = first20["SumbuKecil"].mean()
variansi = first20["SumbuKecil"].std()
display(Math(f'\overline{x} = {{{xbar}}}')')
display(Math(f'\mu_{0} = {{{mean}}}')')
display(Math(f'\sigma = {{{variansi}}}')')
display(Math(f'n = {{{ukuransampel}}}')')
hasil = (xbar-mean)/(variansi/math.sqrt(ukuransampel))
display(Math(f'z = \frac{{{{{xbar}}} - {{{mean}}}}{{{\frac{{{{{variansi}}}}{{{\sqrt{{{{{ukur}}}}}}}}}}}}')')

#Hitung p-value
pvalue = 2*stats.norm.sf(abs(hasil))
display(Math(f'P = {{{pvalue}}}')')

print("\nLangkah 6: Keputusan")
if(pvalue<alpha):
    display(Math(r'\therefore H_{0} \text{{ ditolak karena }} P < \alpha .'))
else:
    display(Math(r'\therefore H_{0} \text{{ tidak ditolak karena }} P > \alpha .'))

#boxplot
sampel2.boxplot("SumbuKecil").set_title("Sampel")
plt.show()
first20.boxplot("SumbuKecil").set_title("Populasi")
plt.show()

```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : \mu = 50$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : \mu \neq 50$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

$$z > 1.96 \text{ atau } z < -1.96$$

$$z = \frac{\overline{x} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$$

Langkah 5: Nilai Uji Statistik

$$\overline{x} = 53.459432142000004$$

$$\mu_0 = 54.887276260499995$$

$$\sigma = 3.373879898276029$$

$$n = 100$$

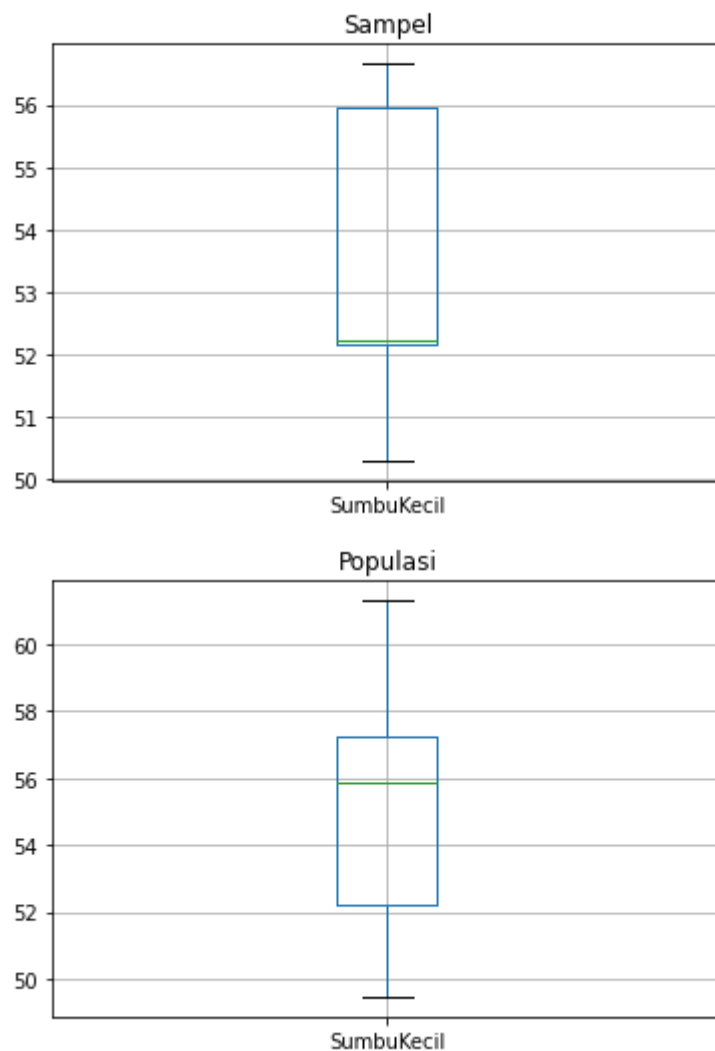
$$z = \frac{53.459432142000004 - 54.887276260499995}{\frac{3.373879898276029}{\sqrt{100}}} = -4.232053782440757$$

$$P = 2.3156701494793453e - 05$$

Langkah 6: Keputusan

$\therefore H_0$  ditolak karena  $P < \alpha$ .





In [35]:

```
#NOMOR 4d: proporsi diameter>85 tidak sama dengan 15%
# Sampel 100 record random
from scipy.stats import binom
alpha = 0.05
df2 = df[df["Diameter"] > 85]
proporsi = len(df2)/len(df)
x = ukuransampel * 0.15

print("\nLangkah 1: Menentukan Hipotesis Nol")
display(Math(r'H_{0}: p = 0.15'))
print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: p \neq 0.15'))
print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
display(Math(f'\alpha = {{alpha}}'))

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
display(Math(f'\text{{Binomial variable X dengan }}p = {{proporsi}} \text{{ dan }} n={{ukuransampel}}'))

print("\nLangkah 5: Nilai Uji Statistik")

#Hitung p-value
pvalue = 2*binom.cdf(x, ukuransampel, proporsi)
display(Math(f'P = {{pvalue}}'))

print("\nLangkah 6: Keputusan")
if(pvalue<alpha):
    display(Math(r'\therefore H_{0} \text{{ ditolak karena }} P < \alpha .'))
else:
    display(Math(r'\therefore H_{0} \text{{ tidak ditolak karena }} P > \alpha .'))

#boxplot
sampel.boxplot("Daerah").set_title("Sampel")
plt.show()
df.boxplot("Daerah").set_title("Populasi")
plt.show()
```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : p = 0.15$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : p \neq 0.15$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

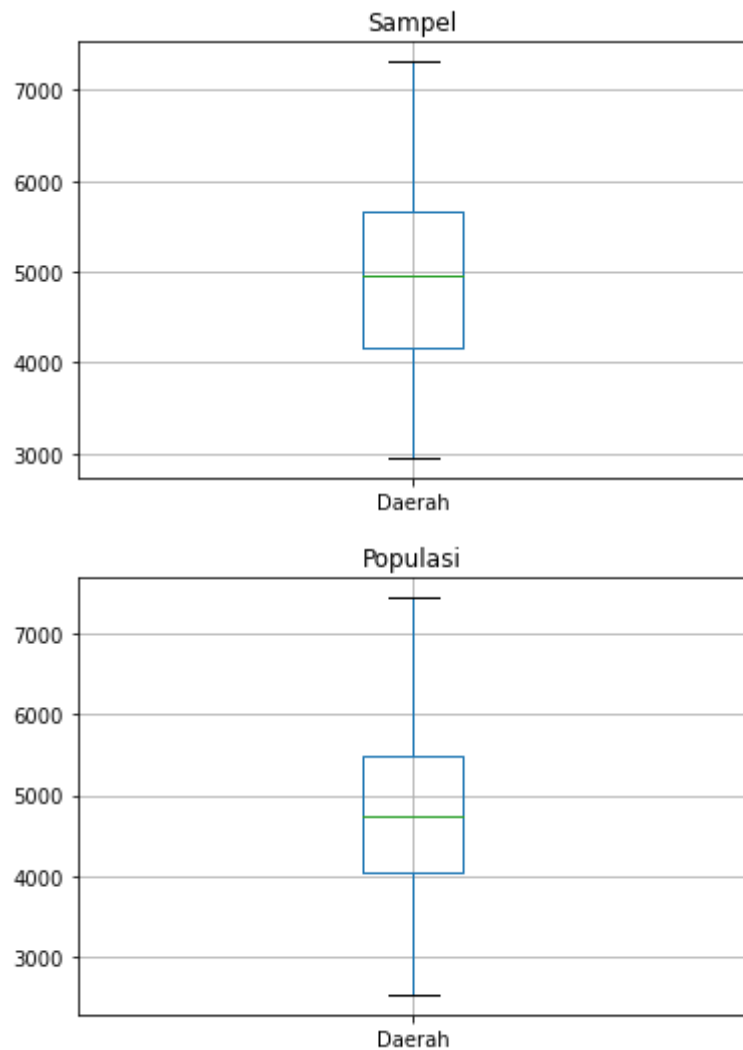
Binomial variable X dengan  $p = 0.194$  dan  $n = 100$

Langkah 5: Nilai Uji Statistik

$$P = 0.323936468785661$$

Langkah 6: Keputusan

$\therefore H_0$  tidak ditolak karena  $P > \alpha$ .



In [36]:

```
#NOMOR 4e: proporsi keliling<100 adalah <5%
df2 = df[df["Keliling"] < 100]
proporsi = len(df2)/len(df)

print("Langkah 1: Menentukan Hipotesis Nol")
display(Math(r'H_{0}: p = 0.05'))
print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: p < 0.05'))
print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
display(Math(f'\alpha = {{{alpha}}}'))

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
z = round(stats.norm.ppf(1-alpha),3)
display(Math(f'z < {{{z}}}'))
display(Math(r'z = \frac{\overline{p}-p_{0}}{\sqrt{\frac{p_{0}q_{0}}{n}}'))

print("\nLangkah 5: Nilai Uji Statistik")
```

```

z = (proporsi-0.05)/math.sqrt(0.05*(1-0.05)/ukuran sampel)
display(Math(f'z = \frac{{{proporsi}}-0.05}{{\sqrt{{\frac{(0.05)(1-0.05)}}{{{ukuran sampel}}}}}}'))

#Hitung p-value
pvalue = stats.norm.sf(abs(z))
display(Math(f'P = {{{pvalue}}}')')

print("\nLangkah 6: Keputusan")
if(pvalue<alpha):
    display(Math(r'\therefore H_{\{0\}} \text{ ditolak karena } P < \alpha .'))
else:
    display(Math(r'\therefore H_{\{0\}} \text{ tidak ditolak karena } P > \alpha .'))

#boxplot
sampel.boxplot("Keliling").set_title("Sampel")
plt.show()
df.boxplot("Keliling").set_title("Populasi")
plt.show()

```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : p = 0.05$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : p < 0.05$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

$$z < 1.645$$

$$z = \frac{\bar{p} - p_0}{\sqrt{\frac{p_0 q_0}{n}}}$$

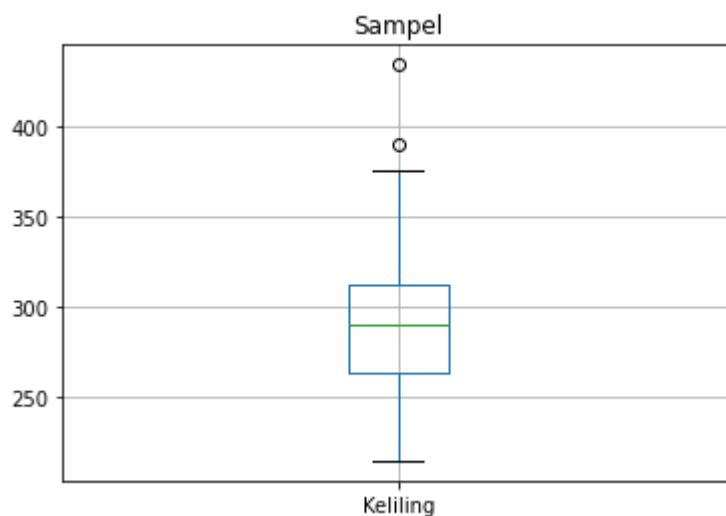
Langkah 5: Nilai Uji Statistik

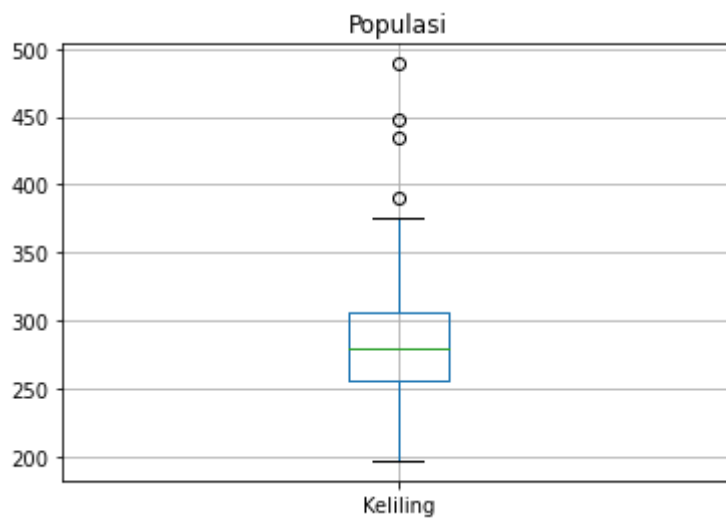
$$z = \frac{0.0 - 0.05}{\sqrt{\frac{(0.05)(1-0.05)}{100}}} = -2.294157338705618$$

$$P = 0.010890731395559738$$

Langkah 6: Keputusan

$\therefore H_0$  ditolak karena  $P < \alpha$ .





## Nomor 5

In [37]:

```
#NOMOR 5a: rata-rata AreaBulatan bagian awal dan akhir sama
dfawal = df.head(250)
dfakhir = df.tail(250)

#Sampel yang diambil adalah 50 record random
ukuransampel = 50
alpha = 0.05
skip1 = sorted(random.sample(range(250), 250-ukuransampel))
skip2 = sorted(random.sample(range(250, 500), 250-ukuransampel))
sampelawal = pd.read_csv('Gandum.csv', skiprows=skip1, delimiter=',', names = ['id', 'Daerah',
sampelawal = sampelawal.head(50)
sampelakhir = pd.read_csv('Gandum.csv', skiprows=skip2, delimiter=',', names = ['id', 'Daerah',
sampelakhir = sampelakhir.tail(50)

print("Langkah 1: Menentukan Hipotesis Nol")
display(Math(r'H_{0}: \mu_{1}-\mu_{2} = 0'))

print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: \mu_{1}-\mu_{2} \neq 0'))

print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
display(Math(f'\alpha = {{alpha}}'))

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
alpha2 = alpha/2
z = round(stats.norm.ppf(1-alpha2), 3)
display(Math(f'z > {{z}} \text{{ atau }} z < {{-z}}'))
display(Math(r'z = \frac{(\overline{x}_{1}-\overline{x}_{2})-d_{0}}{\sqrt{\frac{\sigma_{1}^2}{n_{1}}+\frac{\sigma_{2}^2}{n_{2}}}}'))

print("\nLangkah 5: Nilai Uji Statistik")
xbar1 = sampelawal["AreaBulatan"].mean()
xbar2 = sampelakhir["AreaBulatan"].mean()
meanawal = dfawal["AreaBulatan"].mean()
meanakhir = dfakhir["AreaBulatan"].mean()
variansi1 = dfawal["AreaBulatan"].var()
variansi2 = dfakhir["AreaBulatan"].var()
dnol = 0

display(Math(f'\overline{x}_{1} = {{xbar1}}'))
display(Math(f'\overline{x}_{2} = {{xbar2}}'))
display(Math(f'\mu_{1} = {{meanawal}}'))
display(Math(f'\mu_{2} = {{meanakhir}}'))
display(Math(f'\sigma_{1}^{{2}} = {{variansi1}}'))
display(Math(f'\sigma_{2}^{{2}} = {{variansi2}}'))
display(Math(f'n_{1} = {{ukuransampel}}'))
display(Math(f'n_{2} = {{ukuransampel}}'))

hasil = ((xbar1-xbar2)-dnol)/math.sqrt((variansi1/ukuransampel)+(variansi2/ukuransampel))
```

```
display(Math(f'z = \frac{{{xbar1}}-{{{xbar2}}}-{{{dnol}}}}{{\sqrt{\frac{{{variansi1}}+{{{variansi2}}}}}}'))

#Hitung p-value
pvalue = 2*stats.norm.sf(abs(hasil))
display(Math(f'P = {{{pvalue}}}'))

print("\nLangkah 6: Keputusan")
if(pvalue<alpha):
    display(Math(r'\therefore H_{0} \text{ ditolak karena } P < \alpha .'))
else:
    display(Math(r'\therefore H_{0} \text{ tidak ditolak karena } P > \alpha .'))
```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : \mu_1 - \mu_2 = 0$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : \mu_1 - \mu_2 \neq 0$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

$$z > 1.96 \text{ atau } z < -1.96$$

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - d_0}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Langkah 5: Nilai Uji Statistik

$$\bar{x}_1 = 5374.6$$

$$\bar{x}_2 = 4426.78$$

$$\mu_1 = 5549.804$$

$$\mu_2 = 4324.292$$

$$\sigma_1^2 = 751733.1060080321$$

$$\sigma_2^2 = 545480.4244337347$$

$$n_1 = 50$$

$$n_2 = 50$$

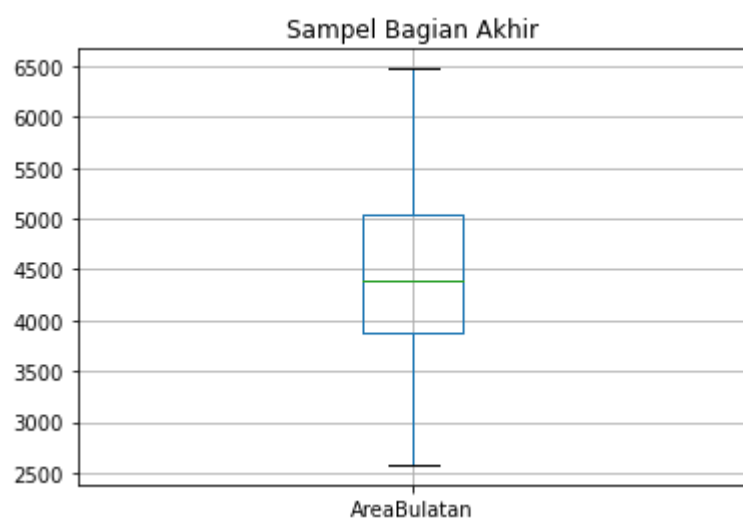
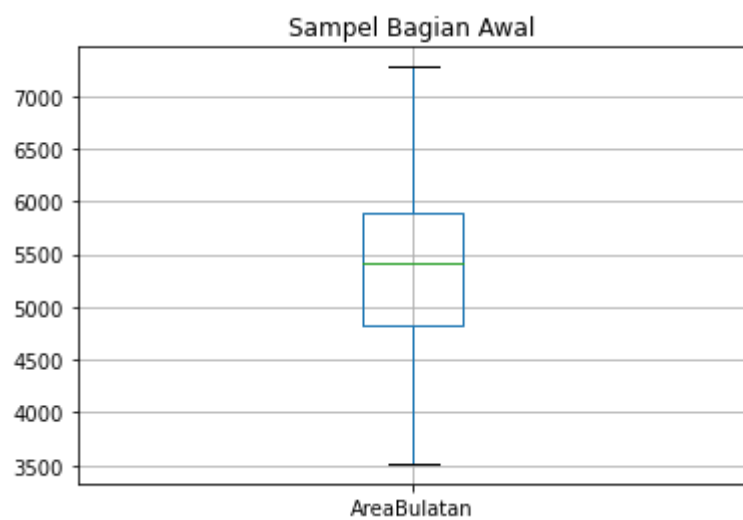
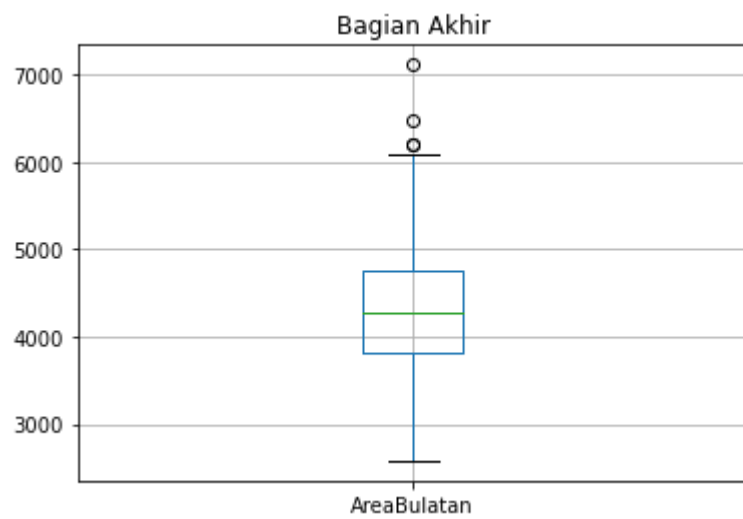
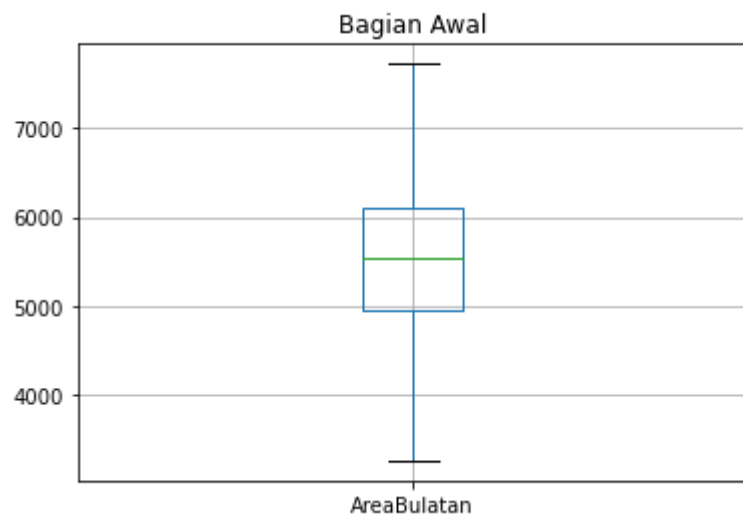
$$z = \frac{(5374.6 - 4426.78) - 0}{\sqrt{\frac{751733.1060080321}{50} + \frac{545480.4244337347}{50}}} = 5.884439956387686$$

$$P = 3.994041003669237e - 09$$

Langkah 6: Keputusan

$\therefore H_0$  ditolak karena  $P < \alpha$ .

```
In [38]: #boxplot
dfawal.boxplot("AreaBulatan").set_title("Bagian Awal")
plt.show()
dfakhir.boxplot("AreaBulatan").set_title("Bagian Akhir")
plt.show()
sampelawal.boxplot("AreaBulatan").set_title("Sampel Bagian Awal")
plt.show()
sampelakhir.boxplot("AreaBulatan").set_title("Sampel Bagian Akhir")
plt.show()
```



In [39]: `#NOMOR 5b: rata-rata KadarAir awal = akhir + 0.2  
print("Langkah 1: Menentukan Hipotesis No1")`

```

display(Math(r'H_{0}: \mu_{1}-\mu_{2} = 0.2'))

print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: \mu_{1}-\mu_{2} \neq 0.2'))

print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
display(Math(f'\alpha = {{{alpha}}}))

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
alpha2 = alpha/2
z = round(stats.norm.ppf(1-alpha2),3)
display(Math(f'z > {{{z}}} \text{ atau } z < {{{-z}}}))
display(Math(r'z = \frac{(\overline{x}_{1}-\overline{x}_{2})-d_{0}}{\sqrt{\frac{\sigma_{1}^2}{n_{1}}+\frac{\sigma_{2}^2}{n_{2}}}}'))

print("\nLangkah 5: Nilai Uji Statistik")
xbar1 = sampelawal["KadarAir"].mean()
xbar2 = sampelakhir["KadarAir"].mean()
meanawal = dfawal["KadarAir"].mean()
meanakhir = dfakhir["KadarAir"].mean()
variansi1 = dfawal["KadarAir"].var()
variansi2 = dfakhir["KadarAir"].var()
dnol = 0.2

display(Math(f'\overline{x}_{1} = {{{xbar1}}}))
display(Math(f'\overline{x}_{2} = {{{xbar2}}}))
display(Math(f'\mu_{1} = {{{meanawal}}}))
display(Math(f'\mu_{2} = {{{meanakhir}}}))
display(Math(f'\sigma_{1}^2 = {{{variansi1}}}))
display(Math(f'\sigma_{2}^2 = {{{variansi2}}}))
display(Math(f'n_{1} = {{{ukuransampel}}}))
display(Math(f'n_{2} = {{{ukuransampel}}}))

hasil = ((xbar1-xbar2)-dnol)/math.sqrt((variansi1/ukuransampel)+(variansi2/ukuransampel))
display(Math(f'z = \frac{({{{{xbar1}}}-{{{xbar2}}})-{{{dnol}}}}{\sqrt{\frac{{{variansi1}}}{n_{1}}+\frac{{{variansi2}}}{n_{2}}}}'))

#Hitung p-value
pvalue = 2*stats.norm.sf(abs(hasil))
display(Math(f'P = {{{pvalue}}}))

print("\nLangkah 6: Keputusan")
if(pvalue<alpha):
    display(Math(r'\therefore H_{0} \text{ ditolak karena } P < \alpha .'))
else:
    display(Math(r'\therefore H_{0} \text{ tidak ditolak karena } P > \alpha .'))

```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : \mu_1 - \mu_2 = 0.2$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : \mu_1 - \mu_2 \neq 0.2$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

$$z > 1.96 \text{ atau } z < -1.96$$

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - d_0}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Langkah 5: Nilai Uji Statistik

$$\bar{x}_1 = 0.6534692853000001$$

$$\bar{x}_2 = 0.6693751876$$

$$\mu_1 = 0.63574344072$$

$$\mu_2 = 0.6609999030760001$$

$$\sigma_1^2 = 0.009043200047076563$$

$$\sigma_2^2 = 0.008482636662870607$$

$$n_1 = 50$$

$$n_2 = 50$$

$$z = \frac{(0.6534692853000001 - 0.6693751876) - 0.2}{\sqrt{\frac{0.009043200047076563}{50} + \frac{0.008482636662870607}{50}}} = -11.532146122057876$$

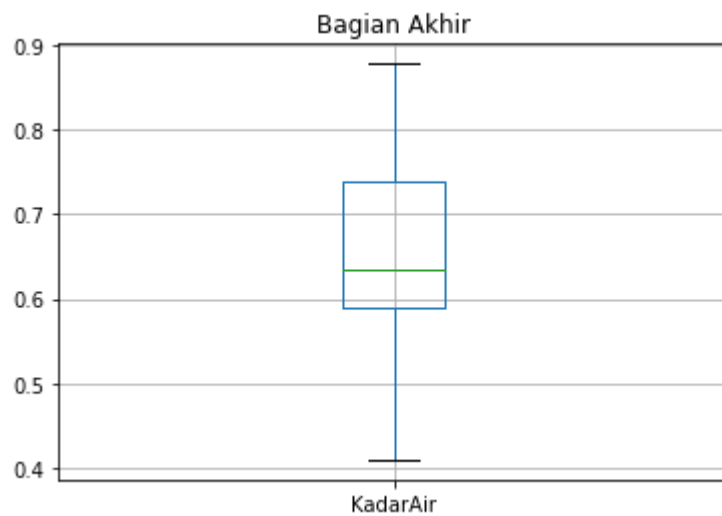
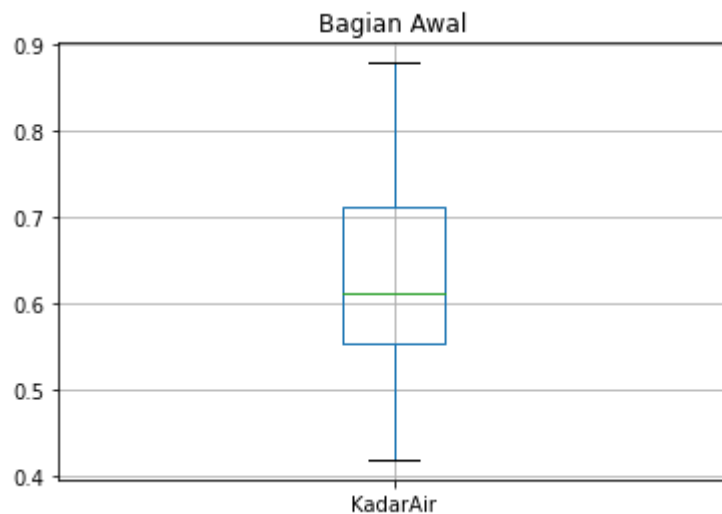
$$P = 9.08503015336961e - 31$$

Langkah 6: Keputusan

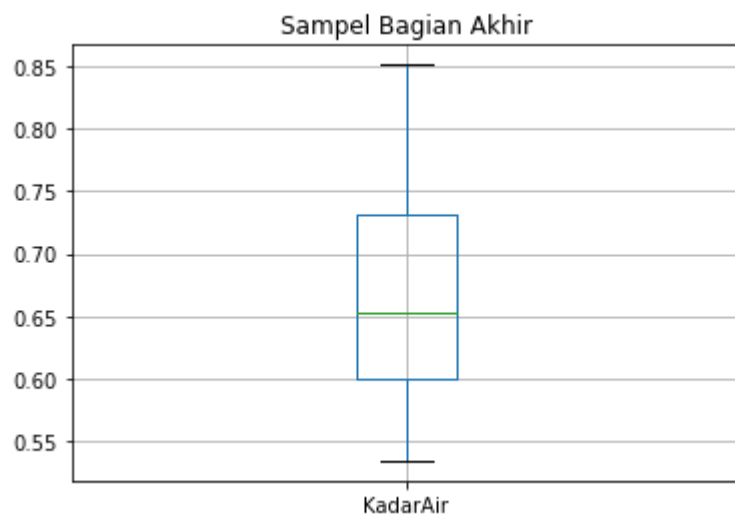
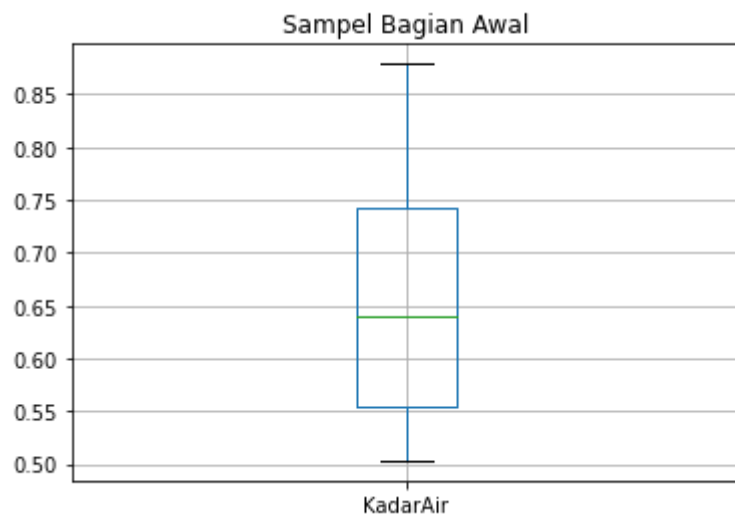
$\therefore H_0$  ditolak karena  $P < \alpha$ .

In [40]:

```
#boxplot
dfawal.boxplot("KadarAir").set_title("Bagian Awal")
plt.show()
dfakhir.boxplot("KadarAir").set_title("Bagian Akhir")
plt.show()
sampelawal.boxplot("KadarAir").set_title("Sampel Bagian Awal")
plt.show()
sampelakhir.boxplot("KadarAir").set_title("Sampel Bagian Akhir")
plt.show()
```







In [41]:

```
#NOMOR 5c: rata-rata 20 baris pertama Bulatan sama dengan 20 baris terakhir
dfawal20 = df.head(20)
dfakhir20 = df.tail(20)

#Sampel yang diambil adalah 50 record random
ukuransampel20 = 5
alpha = 0.05
skip1_20 = sorted(random.sample(range(20),20-ukuransampel20))
skip2_20 = sorted(random.sample(range(480,500),20-ukuransampel20))
sampelawal20 = pd.read_csv('Gandum.csv', skiprows=skip1_20, delimiter=',', names = ['id', 'Daer
sampelawal20 = sampelawal20.head(5)
sampelakhir20 = pd.read_csv('Gandum.csv', skiprows=skip2_20, delimiter=',', names = ['id', 'Daer
sampelakhir20 = sampelakhir20.tail(5)

print("Langkah 1: Menentukan Hipotesis Nol")
display(Math(r'H_{0}: \mu_{1} - \mu_{2} = 0'))
print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: \mu_{1} - \mu_{2} \neq 0'))
print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
display(Math(r'\alpha = 0.05'))

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
alpha2 = alpha/2
z = round(stats.norm.ppf(1-alpha2),3)
display(Math(f'z > {{{z}}} \text{ atau } z < {{{-z}}}'))
display(Math(r'z = \frac{(\overline{x}_{1}-\overline{x}_{2})-d_{0}}{\sqrt{\frac{\sigma_{1}^2}{r

print("\nLangkah 5: Nilai Uji Statistik")

xbar1 = sampelawal20["Bulatan"].mean()
xbar2 = sampelakhir20["Bulatan"].mean()
meanawal = dfawal20["Bulatan"].mean()
meanakhir = dfakhir20["Bulatan"].mean()
variansi1 = dfawal20["Bulatan"].var()
variansi2 = dfakhir20["Bulatan"].var()
```

```

dnol = 0

display(Math(f'\overline{x}_{1} = {{xbar1}}'))
display(Math(f'\overline{x}_{2} = {{xbar2}}'))
display(Math(f'\mu_{1} = {{meanawal}}'))
display(Math(f'\mu_{2} = {{meanakhir}}'))
display(Math(f'\sigma_{1}^{{2}} = {{variansi1}}'))
display(Math(f'\sigma_{2}^{{2}} = {{variansi2}}'))
display(Math(f'n_{1} = {{ukuransampel20}}'))
display(Math(f'n_{2} = {{ukuransampel20}}'))

hasil = ((xbar1-xbar2)-dnol)/math.sqrt((variansi1/ukuransampel)+(variansi2/ukuransampel))
display(Math(f'z = \frac{{{xbar1}}-{{xbar2}}-{{dnol}}}{\sqrt{\frac{{{variansi1}}}{n_1} + \frac{{{variansi2}}}{n_2}}}}'))

#Hitung p-value
pvalue = 2*stats.norm.sf(abs(hasil))
display(Math(f'P = {{pvalue}}'))

print("\nLangkah 6: Keputusan")

display(Math(f'\mu_{1} = {{meanawal}}'))
display(Math(f'\mu_{2} = {{meanakhir}}'))
display(Math(f'\overline{x}_{1} = {{meanawal}}'))
display(Math(f'\overline{x}_{2} = {{meanakhir}}'))

if(pvalue<alpha):
    display(Math(r'\therefore H_{0} \text{ ditolak karena } P < \alpha .'))
else:
    display(Math(r'\therefore H_{0} \text{ tidak ditolak karena } P > \alpha .'))

```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : \mu_1 - \mu_2 = 0$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : \mu_1 - \mu_2 \neq 0$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

$$z > 1.96 \text{ atau } z < -1.96$$

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - d_0}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Langkah 5: Nilai Uji Statistik

$$\bar{x}_1 = 0.7604594432$$

$$\bar{x}_2 = 0.7778442557999999$$

$$\mu_1 = 0.737535352499999$$

$$\mu_2 = 0.767322437$$

$$\sigma_1^2 = 0.0009232346025806985$$

$$\sigma_2^2 = 0.0006307661055759374$$

$$n_1 = 5$$

$$n_2 = 5$$

$$z = \frac{(0.7604594432 - 0.7778442557999999) - 0}{\sqrt{\frac{0.0009232346025806985}{5} + \frac{0.0006307661055759374}{5}}} = -3.1183826792811424$$

$$P = 0.0018184652357509685$$

Langkah 6: Keputusan

$$\mu_1 = 0.7375353552499999$$

$$\mu_2 = 0.767322437$$

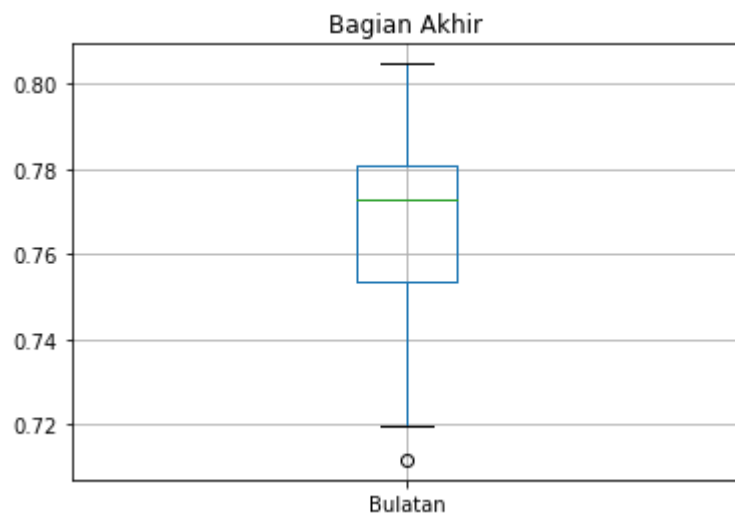
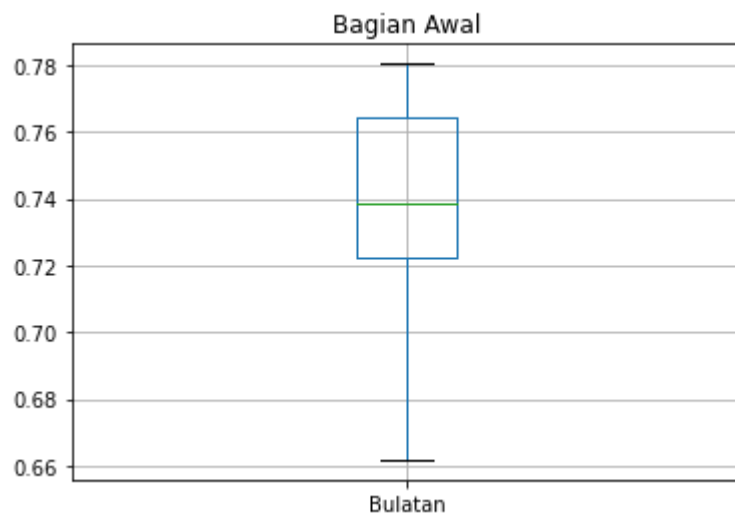
$$\bar{x}_1 = 0.7375353552499999$$

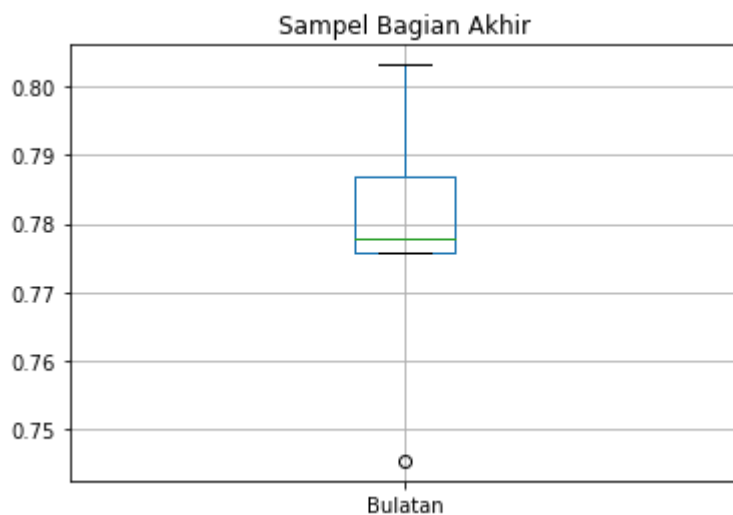
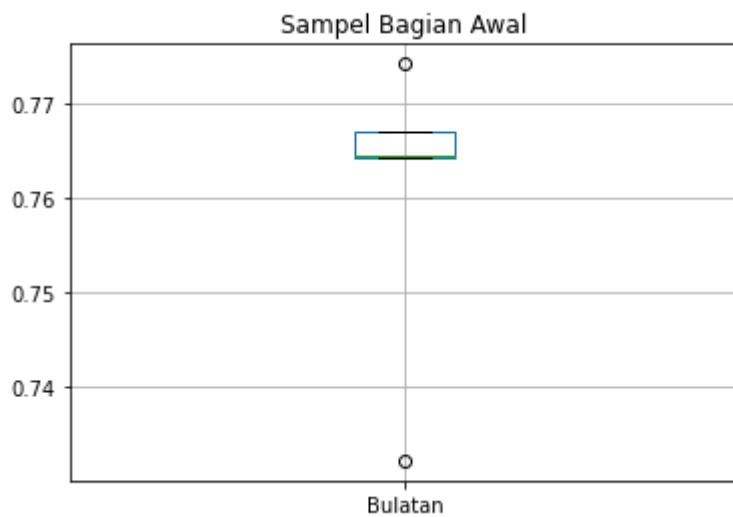
$$\bar{x}_2 = 0.767322437$$

$\therefore H_0$  ditolak karena  $P < \alpha$ .

In [42]:

```
#boxplot
dfawal20.boxplot("Bulatan").set_title("Bagian Awal")
plt.show()
dfakhir20.boxplot("Bulatan").set_title("Bagian Akhir")
plt.show()
sampelawal20.boxplot("Bulatan").set_title("Sampel Bagian Awal")
plt.show()
sampelakhir20.boxplot("Bulatan").set_title("Sampel Bagian Akhir")
plt.show()
```





```
In [43]: #NOMOR 5d: proporsi nilai bagian awal Ransum yang >2 lebih besar dari proporsi nilai yang sama
print("Langkah 1: Menentukan Hipotesis Nol")
display(Math(r'H_{0}: p_{1} = p_{2}'))

print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: p_{1} > p_{2}'))

print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
display(Math(f'\alpha = {{{alpha}}}'))

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
z = round(stats.norm.ppf(1-alpha),3)
display(Math(f'z > {{{z}}}'))
display(Math(r'z = \frac{\overline{p}_{1}-\overline{p}_{2}}{\sqrt{\overline{p}\overline{q}}(\frac{1}{n_{1}}+\frac{1}{n_{2}})})')

print("\nLangkah 5: Nilai Uji Statistik")
df3 = df[df["Ransum"]>2]
proporsi = len(df3)/len(df)
q = 1 - proporsi
psampelawal = dfawal[dfawal["Ransum"]>2]
psampelakhir = dfakhir[dfakhir["Ransum"]>2]
p1 = len(psampelawal)/len(dfawal)
p2 = len(psampelakhir)/len(dfakhir)

display(Math(f'\overline{p} = {{{proporsi}}}'))
display(Math(f'\overline{q} = {{{q}}}'))
display(Math(f'\overline{p}_{1} = {{{p1}}}'))
display(Math(f'\overline{p}_{2} = {{{p2}}}'))
display(Math(f'n_{1} = {{{ukuransampel}}}'))
display(Math(f'n_{2} = {{{ukuransampel}}}'))

hasil = (p1-p2)/(math.sqrt(proporsi*q*(1/ukuransampel+1/ukuransampel)))
display(Math(f'z = {{{hasil}}}'))

#Hitung p-value
```

```

pvalue = stats.norm.sf(abs(hasil))
display(Math(f'P = {{{pvalue}}}')

print("\nLangkah 6: Keputusan")
if(pvalue<alpha):
    display(Math(r'\text{ therefore } H_{0} \text{ ditolak karena } P < \alpha .'))
else:
    display(Math(r'\text{ therefore } H_{0} \text{ tidak ditolak karena } P > \alpha .'))

```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : p_1 = p_2$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : p_1 > p_2$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

$$z > 1.645$$

$$z = \frac{\bar{p}_1 - \bar{p}_2}{\sqrt{\bar{p}\bar{q}\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

Langkah 5: Nilai Uji Statistik

$$\bar{p} = 0.73$$

$$\bar{q} = 0.27$$

$$\bar{p}_1 = 0.996$$

$$\bar{p}_2 = 0.464$$

$$n_1 = 50$$

$$n_2 = 50$$

$$z = 5.991538088475442$$

$$P = 1.039328134043544e - 09$$

Langkah 6: Keputusan

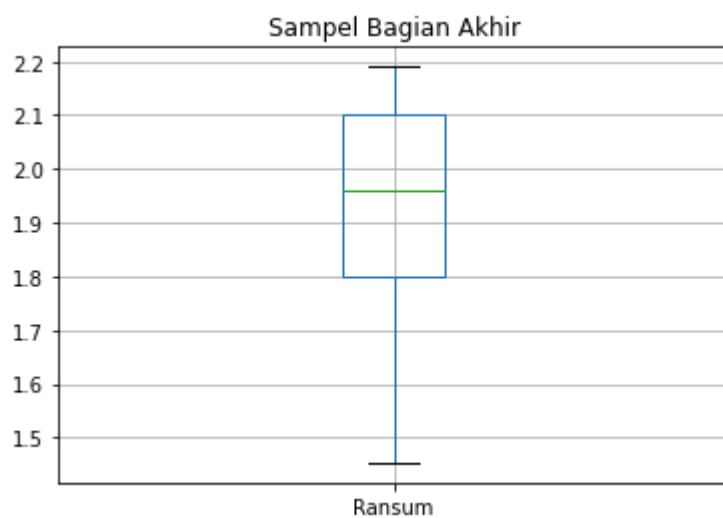
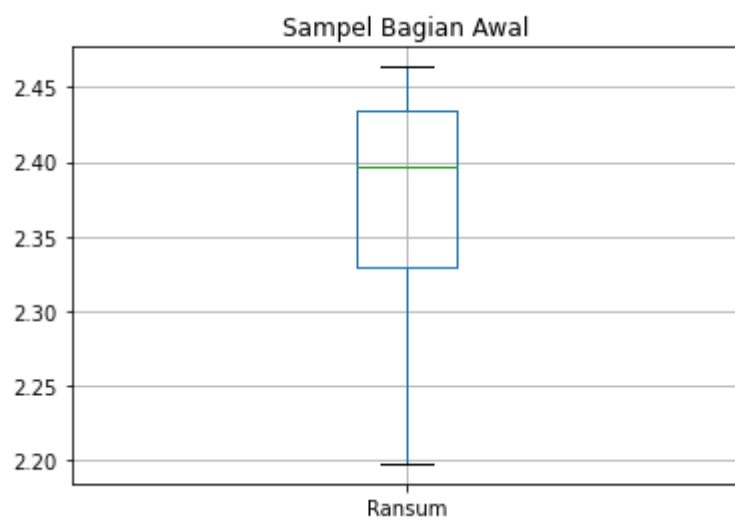
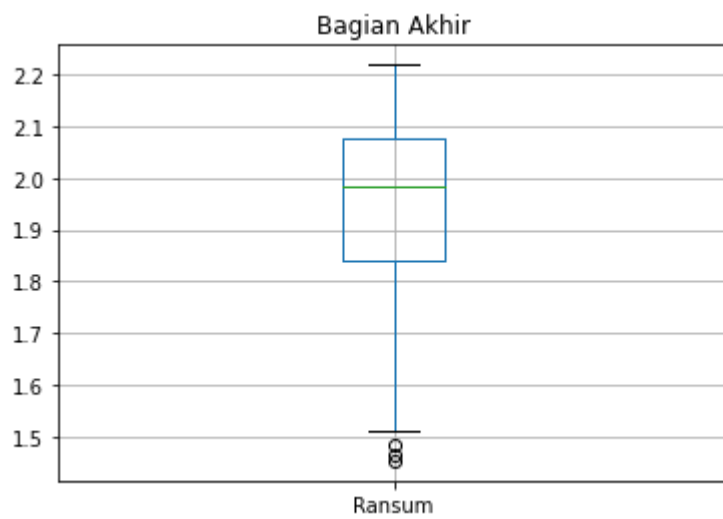
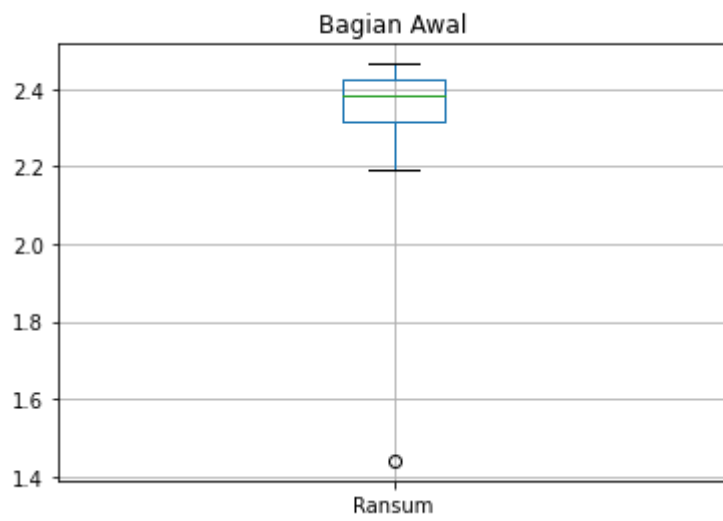
$\therefore H_0$  ditolak karena  $P < \alpha$ .

In [44]:

```

#boxplot
dfawal.boxplot("Ransum").set_title("Bagian Awal")
plt.show()
dfakhir.boxplot("Ransum").set_title("Bagian Akhir")
plt.show()
sampelawal.boxplot("Ransum").set_title("Sampel Bagian Awal")
plt.show()
sampelakhir.boxplot("Ransum").set_title("Sampel Bagian Akhir")
plt.show()

```



In [57]: `#NOMOR 5e: bagian awal Diameter variansinya sama dengan bagian akhir  
from scipy.stats import f`

```

alpha = 0.05
print("Langkah 1: Menentukan Hipotesis Nol")
display(Math(r'H_{0}: \sigma_{1}^2 = \sigma_{2}^2'))

print("\nLangkah 2: Menentukan Hipotesis Alternatif")
display(Math(r'H_{1}: \sigma_{1}^2 \neq \sigma_{2}^2'))

print("\nLangkah 3: Menentukan Tingkat Signifikan Alpha")
display(Math(f'\alpha = {{{alpha}}}))

print("\nLangkah 4: Menentukan Uji Statistik dan Daerah Kritis")
alpha2 = alpha/2
f1 = round(f.ppf(alpha2, 49, 49),3)
f2 = round(1/f1,3)
display(Math(f'f < {{{f1}}} \text{{ atau }} f > {{{f2}}}))
#belum
display(Math(r'f = \frac{s_{1}^2}{s_{2}^2}'))

print("\nLangkah 5: Nilai Uji Statistik")
s1 = sampelawal["Diameter"].var()
s2 = sampelakhir["Diameter"].var()

display(Math(f's_{1}^2 = {{{s1}}}))
display(Math(f's_{2}^2 = {{{s2}}}))
hasil = s1/s2
display(Math(f'f = {{{hasil}}}))

#hitung p-value
pvalue = 1-f.cdf(hasil, 49, 49)
display(Math(f'P = {{{pvalue}}}))

print("\nLangkah 6: Keputusan")
if(pvalue<alpha):
    display(Math(r'\therefore H_{0} \text{{ ditolak karena }} P < \alpha .'))
else:
    display(Math(r'\therefore H_{0} \text{{ tidak ditolak karena }} P > \alpha .'))

```

Langkah 1: Menentukan Hipotesis Nol

$$H_0 : \sigma_1^2 = \sigma_2^2$$

Langkah 2: Menentukan Hipotesis Alternatif

$$H_1 : \sigma_1^2 \neq \sigma_2^2$$

Langkah 3: Menentukan Tingkat Signifikan Alpha

$$\alpha = 0.05$$

Langkah 4: Menentukan Uji Statistik dan Daerah Kritis

$$f < 0.567 \text{ atau } f > 1.764$$

$$f = \frac{s_1^2}{s_2^2}$$

Langkah 5: Nilai Uji Statistik

$$s_1^2 = 48.67296764900209$$

$$s_2^2 = 58.50327955806157$$

$$f = 0.831969900092466$$

$$P = 0.7389565179905937$$

Langkah 6: Keputusan

$\therefore H_0$  tidak ditolak karena  $P > \alpha$ .

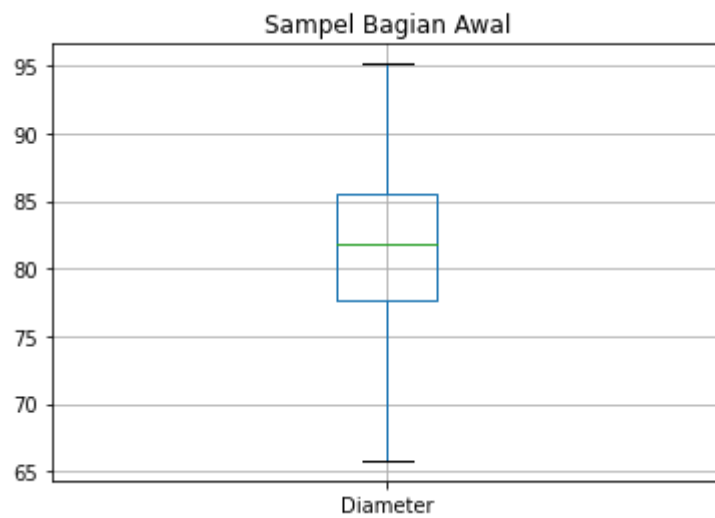
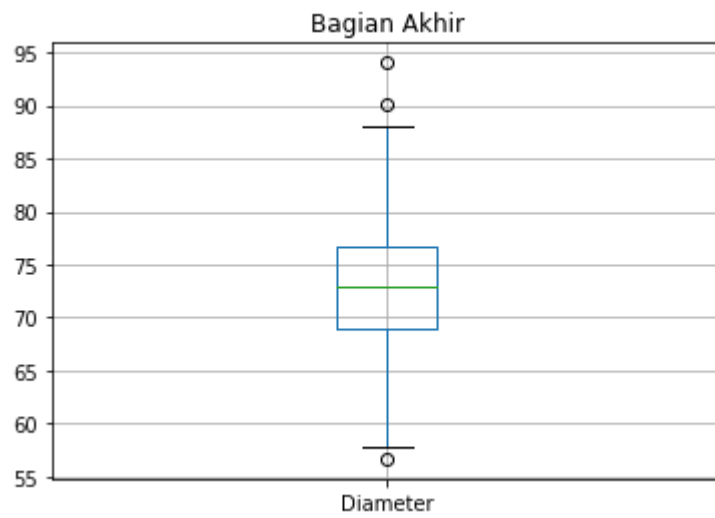
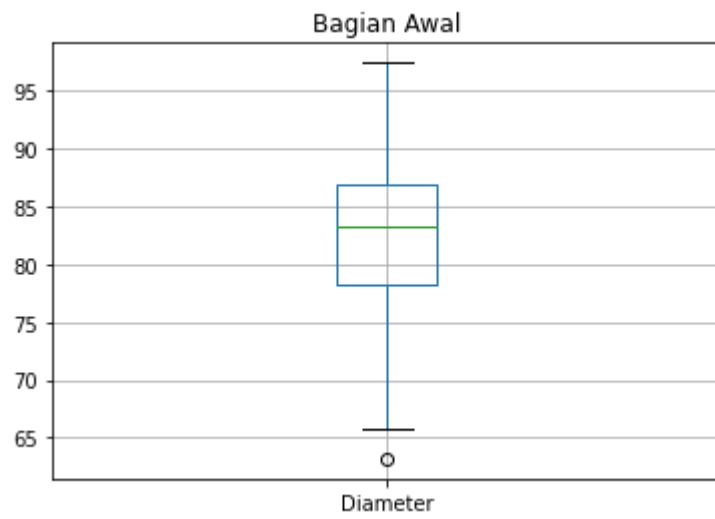
In [58]:

```

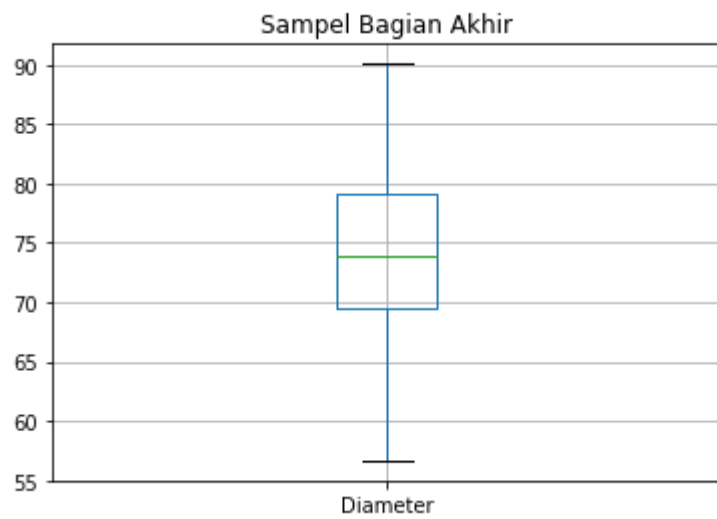
#boxplot
dfawal.boxplot("Diameter").set_title("Bagian Awal")
plt.show()
dfakhir.boxplot("Diameter").set_title("Bagian Akhir")
plt.show()

```

```
sampelawal.boxplot("Diameter").set_title("Sampel Bagian Awal")  
plt.show()  
sampelakhir.boxplot("Diameter").set_title("Sampel Bagian Akhir")  
plt.show()
```







## Nomor 6

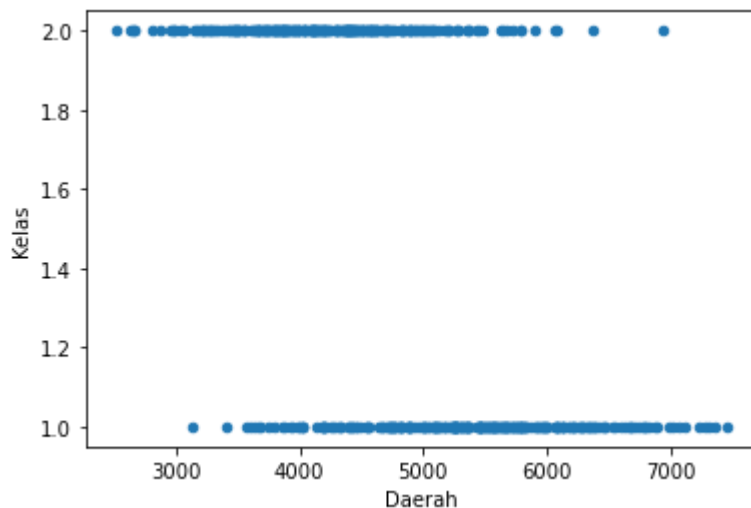
### Korelasi Daerah

```
In [46]: #NOMOR 6
inpkor = "Daerah"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

if(hasilkor>0):
    print("Korelasi berhubungan")
else:
    print("Korelasi tidak berhubungan")
df.plot(kind='scatter',x=inpkor,y='Kelas')
```

Korelasi antara Daerah dan Kelas :  
 -0.6027466517416654  
 Korelasi tidak berhubungan

Out[46]: <AxesSubplot:xlabel='Daerah', ylabel='Kelas'>



### Korelasi SumbuUtama

```
In [47]: inpkor = "SumbuUtama"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

if(hasilkor>0):
    print("Korelasi berhubungan")
else:
```

```
print("Korelasi tidak berhubungan")
```

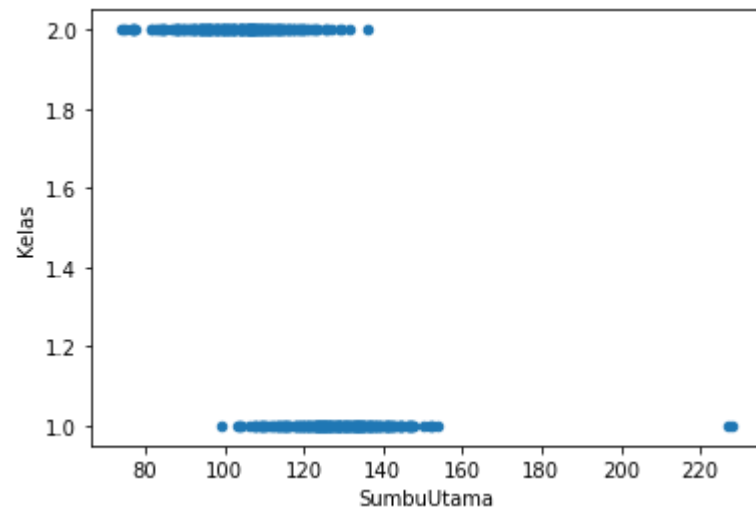
```
df.plot(kind='scatter',x=inpkor,y='Kelas')
```

Korelasi antara SumbuUtama dan Kelas :

-0.7130906104204592

Korelasi tidak berhubungan

Out[47]: <AxesSubplot:xlabel='SumbuUtama', ylabel='Kelas'>



## Korelasi SumbuKecil

```
In [48]: inpkor = "SumbuKecil"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

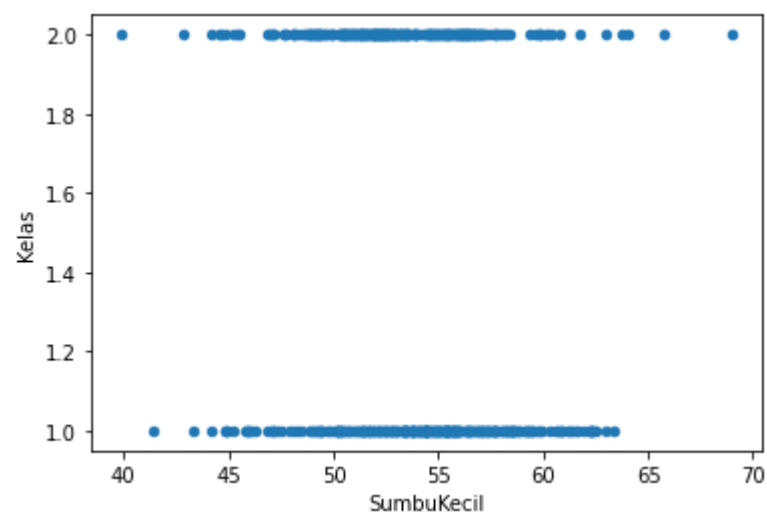
if(hasilkor>0):
    print("Korelasi berhubungan")
else:
    print("Korelasi tidak berhubungan")
df.plot(kind='scatter',x=inpkor,y='Kelas')
```

Korelasi antara SumbuKecil dan Kelas :

-0.15297517335535024

Korelasi tidak berhubungan

Out[48]: <AxesSubplot:xlabel='SumbuKecil', ylabel='Kelas'>



## Korelasi Keunikan

```
In [49]: inpkor = "Keunikan"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

if(hasilkor>0):
    print("Korelasi berhubungan")
```

```

else:
    print("Korelasi tidak berhubungan")
df.plot(kind='scatter',x=inpkor,y='Kelas')

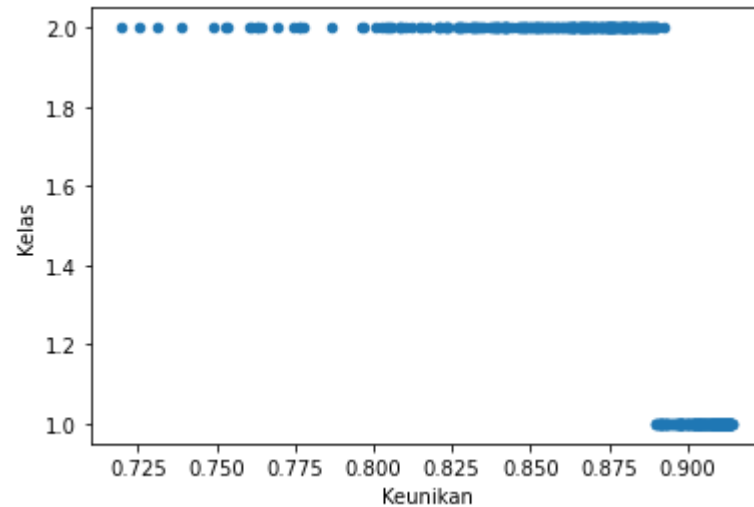
```

Korelasi antara Keunikan dan Kelas :

-0.7304563686511927

Korelasi tidak berhubungan

Out[49]: <AxesSubplot:xlabel='Keunikan', ylabel='Kelas'>



## Korelasi AreaBulatan

```

In [50]: inpkor = "AreaBulatan"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

if(hasilkor>0):
    print("Korelasi berhubungan")
else:
    print("Korelasi tidak berhubungan")
df.plot(kind='scatter',x=inpkor,y='Kelas')

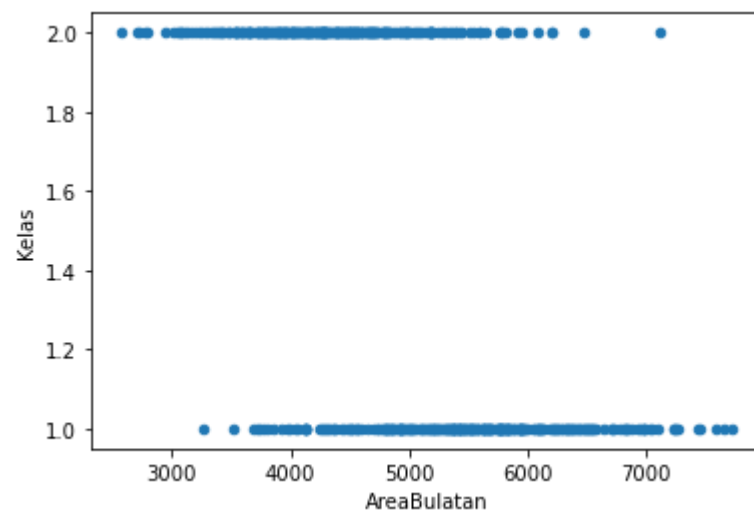
```

Korelasi antara AreaBulatan dan Kelas :

-0.6073125434153747

Korelasi tidak berhubungan

Out[50]: <AxesSubplot:xlabel='AreaBulatan', ylabel='Kelas'>



## Korelasi Diameter

```

In [51]: inpkor = "Diameter"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

if(hasilkor>0):

```

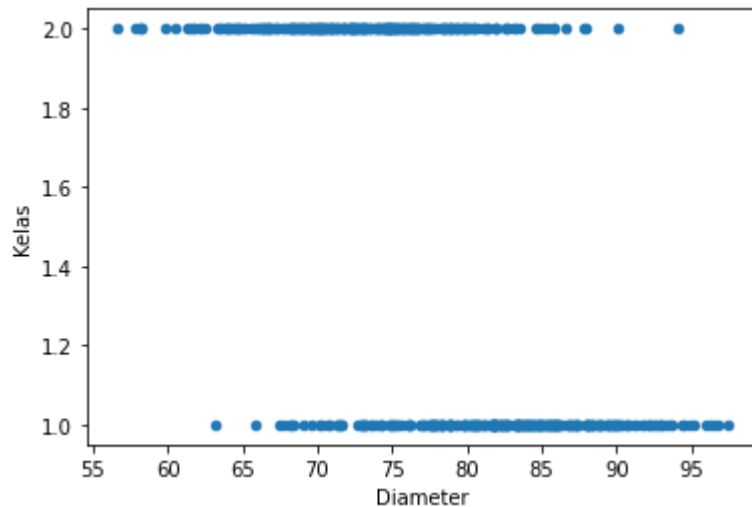
```

print("Korelasi berhubungan")
else:
    print("Korelasi tidak berhubungan")
df.plot(kind='scatter',x=inpkor,y='Kelas')

```

Korelasi antara Diameter dan Kelas :  
-0.6025356896618811  
Korelasi tidak berhubungan

Out[51]: <AxesSubplot:xlabel='Diameter', ylabel='Kelas'>



## Korelasi KadarAir

```

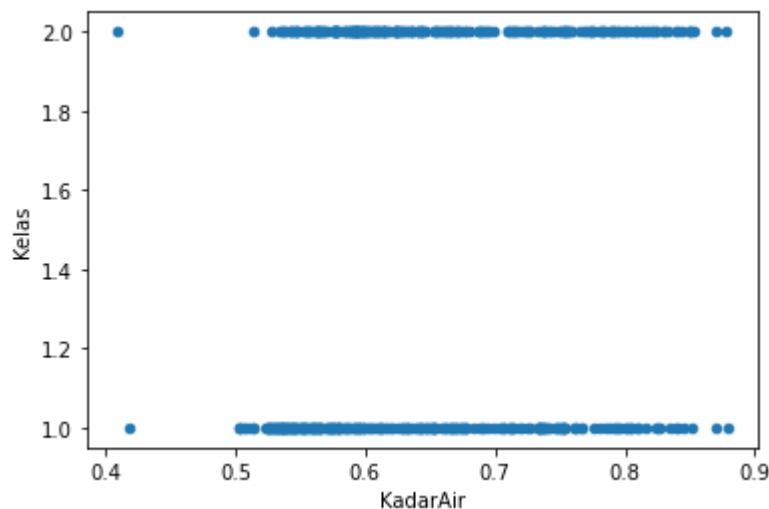
In [52]: inpkor = "KadarAir"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

if(hasilkor>0):
    print("Korelasi berhubungan")
else:
    print("Korelasi tidak berhubungan")
df.plot(kind='scatter',x=inpkor,y='Kelas')

```

Korelasi antara KadarAir dan Kelas :  
0.13434422605727642  
Korelasi berhubungan

Out[52]: <AxesSubplot:xlabel='KadarAir', ylabel='Kelas'>



## Korelasi Keliling

```

In [53]: inpkor = "Keliling"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

```

```

if(hasilkor>0):
    print("Korelasi berhubungan")
else:
    print("Korelasi tidak berhubungan")
df.plot(kind='scatter',x=inpkor,y='Kelas')

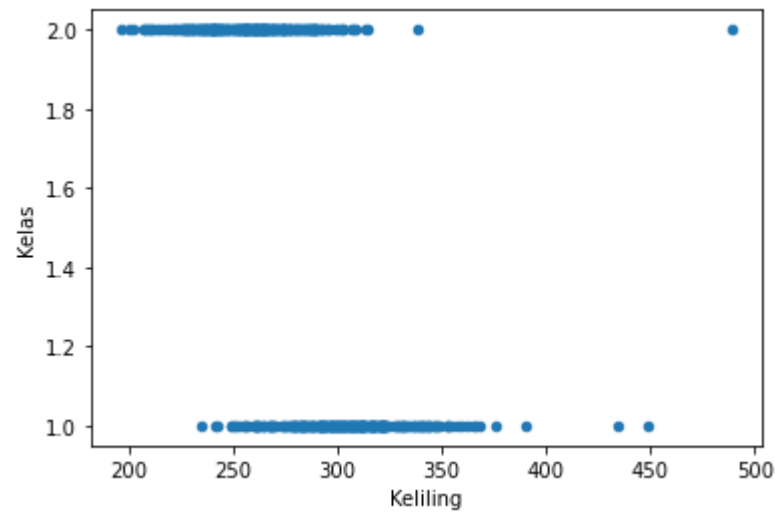
```

Korelasi antara Keliling dan Kelas :

-0.6348607454756853

Korelasi tidak berhubungan

Out[53]: <AxesSubplot:xlabel='Keliling', ylabel='Kelas'>



## Korelasi Bulatan

```

In [54]: inpkor = "Bulatan"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

if(hasilkor>0):
    print("Korelasi berhubungan")
else:
    print("Korelasi tidak berhubungan")
df.plot(kind='scatter',x=inpkor,y='Kelas')

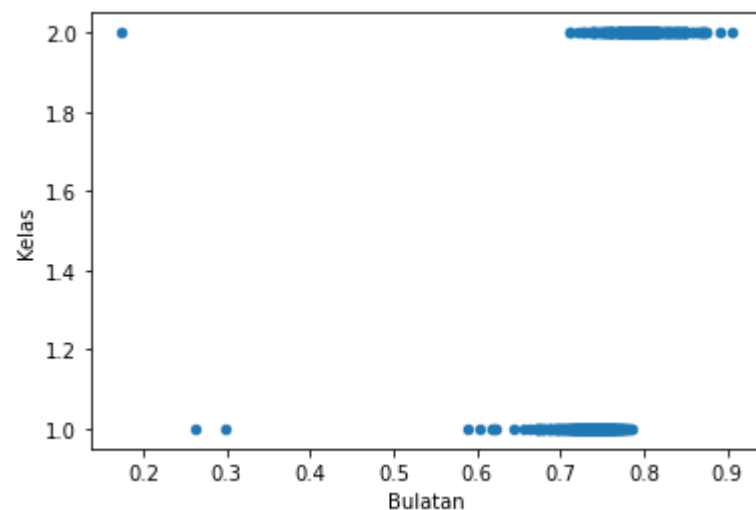
```

Korelasi antara Bulatan dan Kelas :

0.5450045317240073

Korelasi berhubungan

Out[54]: <AxesSubplot:xlabel='Bulatan', ylabel='Kelas'>



## Korelasi Ransum

```

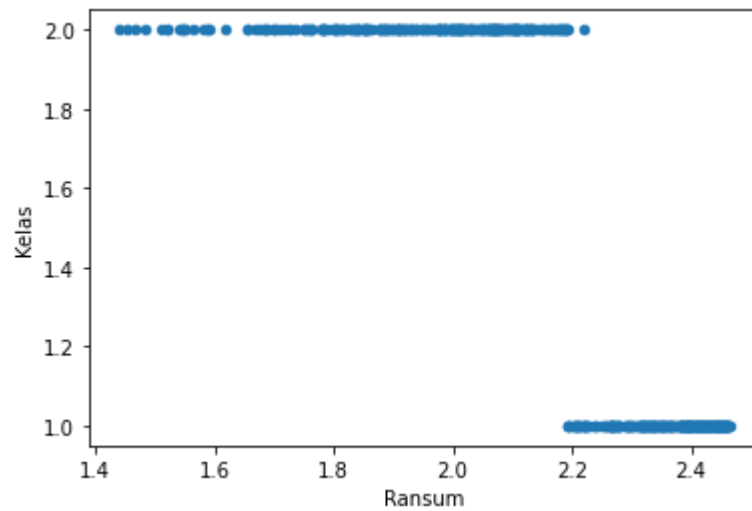
In [55]: inpkor = "Ransum"
print("Korelasi antara",str(inpkor), "dan Kelas :")
hasilkor = df[inpkor].corr(df["Kelas"])
print(hasilkor)

```

```
if(hasilkor>0):  
    print("Korelasi berhubungan")  
else:  
    print("Korelasi tidak berhubungan")  
df.plot(kind='scatter',x=inpkor,y='Kelas')
```

Korelasi antara Ransum dan Kelas :  
-0.8399038681287486  
Korelasi tidak berhubungan

Out[55]: <AxesSubplot:xlabel='Ransum', ylabel='Kelas'>



In [ ]: