

probbat

April 16, 2021

```
[1]: import pandas as pd
import pylab
import scipy.stats as stats
import math

print("Anggota: ")
print("Ilyasa Salafi Putra Jamal 13519023")
print("Mahameru Ds 13519014")

df = pd.read_csv("Gandum.csv")
```

Anggota:

Ilyasa Salafi Putra Jamal 13519023

```
[2]: df.describe()
```

```
[2]:
```

	id	Daerah	SumbuUtama	SumbuKecil	Keunikan	\
count	500.000000	500.000000	500.000000	500.000000	500.000000	
mean	250.500000	4801.246000	116.045171	53.715246	0.878764	
std	144.481833	986.395491	18.282626	4.071075	0.036586	
min	1.000000	2522.000000	74.133114	39.906517	0.719916	
25%	125.750000	4042.750000	104.116098	51.193576	0.863676	
50%	250.500000	4735.000000	115.405140	53.731199	0.890045	
75%	375.250000	5495.500000	129.046792	56.325158	0.907578	
max	500.000000	7453.000000	227.928583	68.977700	0.914001	

	AreaBulatan	Diameter	KadarAir	Keliling	Bulatan	\
count	500.000000	500.000000	500.000000	500.000000	500.000000	
mean	4937.048000	77.771158	0.648372	281.479722	0.761737	
std	1011.696255	8.056867	0.094367	37.335402	0.061702	
min	2579.000000	56.666658	0.409927	197.015000	0.174590	
25%	4170.250000	71.745308	0.572632	255.883000	0.731991	
50%	4857.000000	77.645277	0.626117	280.045500	0.761288	
75%	5654.250000	83.648598	0.726633	306.062500	0.796361	
max	7720.000000	97.413830	0.878899	488.837000	0.904748	

	Ransum	Kelas
count	500.000000	500.000000

mean	2.150915	1.502000
std	0.249767	0.500497
min	1.440796	1.000000
25%	1.983939	1.000000
50%	2.193599	2.000000
75%	2.381612	2.000000
max	2.464809	2.000000

```
[3]: print("Skewness: ")
      df.skew()
```

Skewness:

```
[3]: id          0.000000
      Daerah      0.238144
      SumbuUtama  0.761529
      SumbuKecil -0.010828
      Keunikan    -1.623472
      AreaBulatan 0.257560
      Diameter    0.002725
      KadarAir    0.493661
      Keliling    0.733627
      Bulatan     -3.599237
      Ransum      -0.658188
      Kelas       -0.008024
      dtype: float64
```

```
[4]: print("Variance: ")
      df.var()
```

Variance:

```
[4]: id          2.087500e+04
      Daerah      9.729761e+05
      SumbuUtama  3.342544e+02
      SumbuKecil  1.657365e+01
      Keunikan    1.338528e-03
      AreaBulatan 1.023529e+06
      Diameter    6.491311e+01
      KadarAir    8.905149e-03
      Keliling    1.393932e+03
      Bulatan     3.807194e-03
      Ransum      6.238350e-02
      Kelas       2.504970e-01
      dtype: float64
```

```
[5]: print("Kurtosis: ")
      df.kurt()
```

Kurtosis:

```
[5]: id          -1.200000
     Daerah      -0.434631
     SumbuUtama   4.330534
     SumbuKecil   0.475568
     Keunikan     2.917256
     AreaBulatan  -0.409685
     Diameter     -0.466455
     KadarAir     -0.740326
     Keliling     2.272685
     Bulatan      29.975096
     Ransum       -0.428656
     Kelas        -2.007984
     dtype: float64
```

```
[6]: print("Modus: ")
     print("Beberapa data memiliki 500 modus karena setiap datanya unik")
     df.mode()
```

Modus:

Beberapa data memiliki 500 modus karena setiap datanya unik

```
[6]:
```

	id	Daerah	SumbuUtama	SumbuKecil	Keunikan	AreaBulatan	Diameter \
0	1	3992.0	74.133114	39.906517	0.719916	3802.0	71.293564
1	2	4881.0	74.364021	41.436419	0.725553	4913.0	78.833256
2	3	5642.0	74.691881	42.871879	0.731211	NaN	84.756224
3	4	6083.0	76.293164	43.284979	0.738639	NaN	88.006342
4	5	NaN	76.789043	44.119355	0.749282	NaN	NaN
..
495	496	NaN	152.068440	63.322854	0.913724	NaN	NaN
496	497	NaN	152.113491	63.762307	0.913760	NaN	NaN
497	498	NaN	153.583387	64.012769	0.913891	NaN	NaN
498	499	NaN	227.105462	65.738475	0.913909	NaN	NaN
499	500	NaN	227.928583	68.977700	0.914001	NaN	NaN

	KadarAir	Keliling	Bulatan	Ransum	Kelas
0	0.735849	197.015	0.174590	1.440796	2.0
1	0.824405	200.587	0.261297	1.453137	NaN
2	NaN	202.456	0.299298	1.465950	NaN
3	NaN	207.325	0.589146	1.483456	NaN
4	NaN	207.697	0.603807	1.510000	NaN
..
495	NaN	375.651	0.872417	2.461017	NaN
496	NaN	390.125	0.874243	2.461510	NaN
497	NaN	434.235	0.874743	2.463297	NaN
498	NaN	448.305	0.891706	2.463546	NaN
499	NaN	488.837	0.904748	2.464809	NaN

[500 rows x 12 columns]

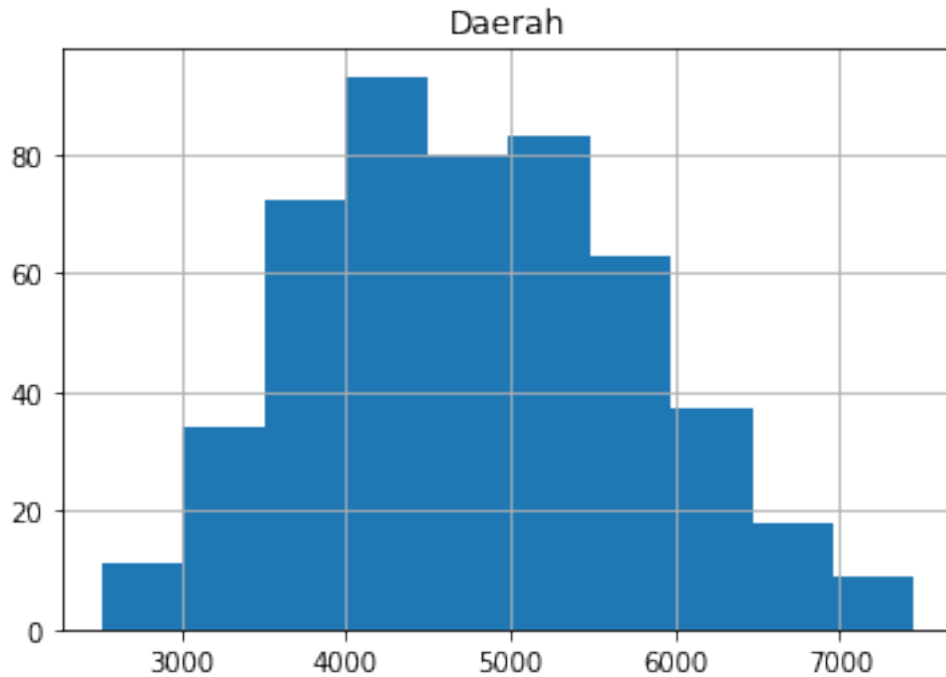
```
[7]: print("Interquartile Range (IQR):")
val_list = df["Daerah"].values.tolist()
iqr = stats.iqr(val_list)
print("Daerah\t\t: ", iqr)
val_list = df["SumbuUtama"].values.tolist()
iqr = stats.iqr(val_list)
print("Sumbu Utama\t: ", iqr)
val_list = df["SumbuKecil"].values.tolist()
iqr = stats.iqr(val_list)
print("Sumbu Kecil\t: ", iqr)
val_list = df["Keunikan"].values.tolist()
iqr = stats.iqr(val_list)
print("Keunikan\t: ", iqr)
val_list = df["AreaBulatan"].values.tolist()
iqr = stats.iqr(val_list)
print("Area Bulatan\t: ", iqr)
val_list = df["Diameter"].values.tolist()
iqr = stats.iqr(val_list)
print("Diameter\t: ", iqr)
val_list = df["KadarAir"].values.tolist()
iqr = stats.iqr(val_list)
print("Kadar Air\t: ", iqr)
val_list = df["Keliling"].values.tolist()
iqr = stats.iqr(val_list)
print("Keliling\t: ", iqr)
val_list = df["Bulatan"].values.tolist()
iqr = stats.iqr(val_list)
print("Bulatan\t\t\t: ", iqr)
val_list = df["Ransum"].values.tolist()
iqr = stats.iqr(val_list)
print("Ransum\t\t: ", iqr)
val_list = df["Kelas"].values.tolist()
iqr = stats.iqr(val_list)
print("Kelas\t\t\t: ", iqr)
```

```
Interquartile Range (IQR):
Daerah      : 1452.75
Sumbu Utama : 24.930693850000001
Sumbu Kecil : 5.1315816500000001
Keunikan    : 0.04390216424999993
Area Bulatan : 1484.0
Diameter    : 11.903290020000014
Kadar Air   : 0.15400088725000005
Keliling    : 50.179500000000002
Bulatan     : 0.06437024175000006
```

```
Ransum      : 0.3976734275  
Kelas      : 1.0
```

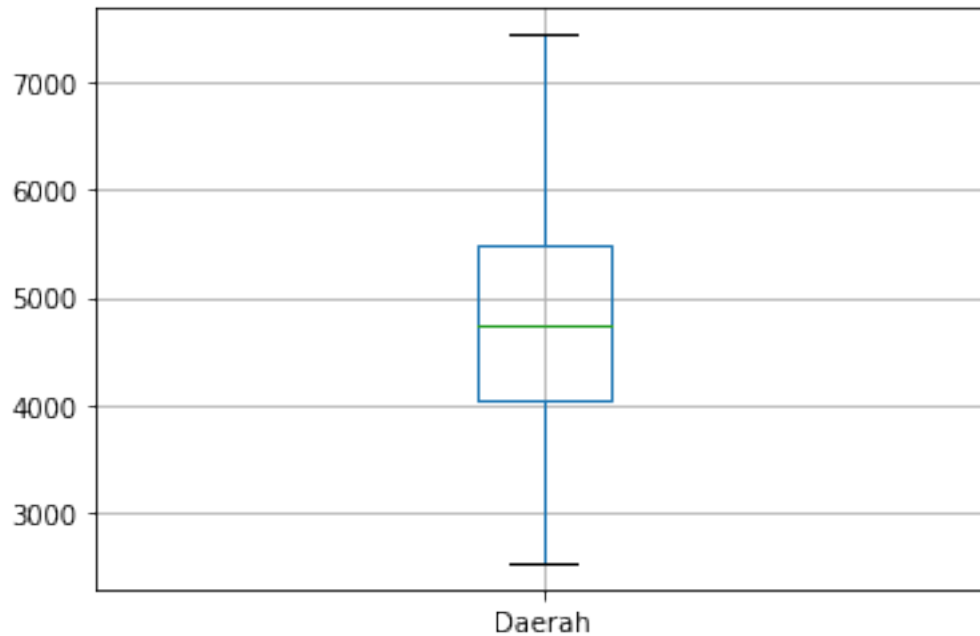
```
[8]: df.hist(column="Daerah")
```

```
[8]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC7A6A948>]],  
      dtype=object)
```



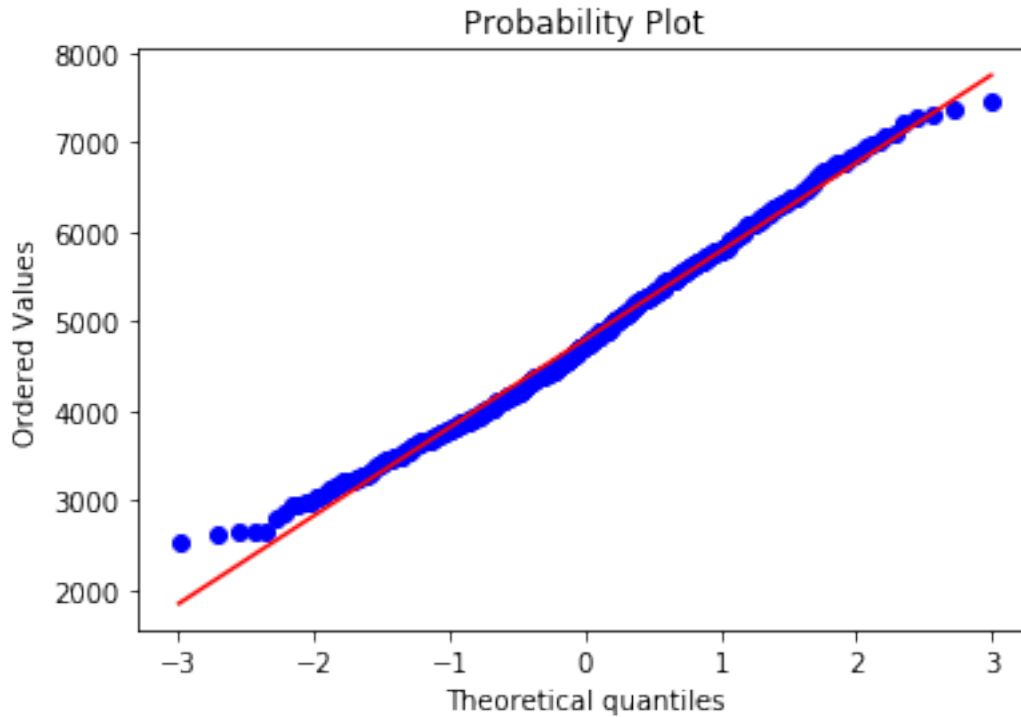
```
[9]: df.boxplot(column="Daerah")
```

```
[9]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac8203dc8>
```



```
[10]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
      ↪ ")
stats.probplot(df["Daerah"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk")
```

Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk

```
[11]: print("Pengujian normal dengan tes Shapiro-Wilk")
      val_list = df["Daerah"].values.tolist()
      tset, pval = stats.shapiro(val_list)
      print("Pval: ", pval)
      if pval > 0.05:
          print("Pval lebih besar dari 0.05, Kesimpulan: Berdistribusi normal")
      else:
          print("Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal")
      ↪ #Ans
```

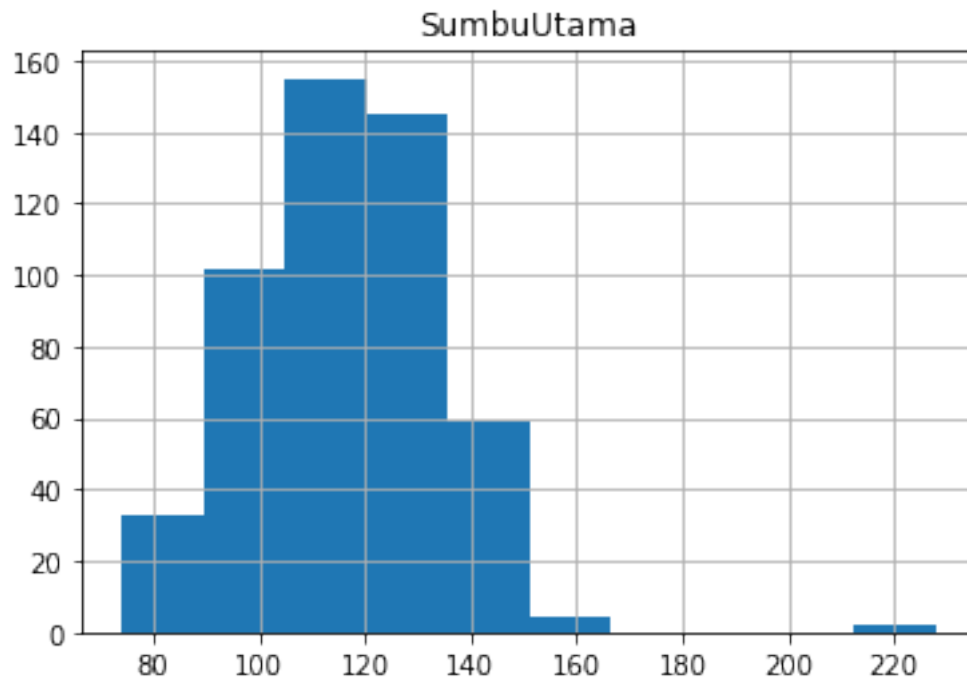
Pengujian normal dengan tes Shapiro-Wilk

Pval: 0.003270698245614767

Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal

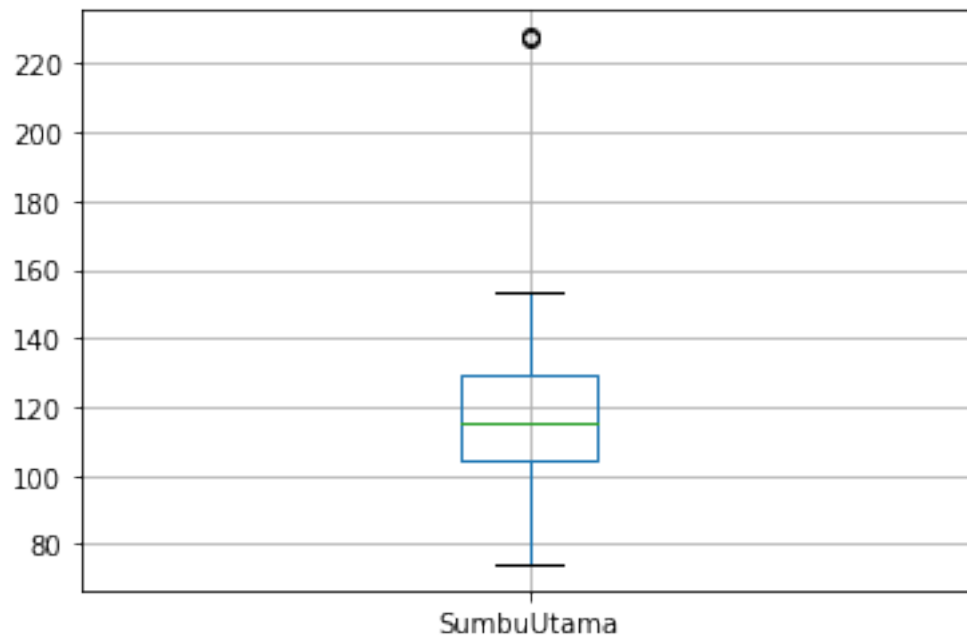
```
[12]: df.hist(column="SumbuUtama")
```

```
[12]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC828F288>]],
      dtype=object)
```



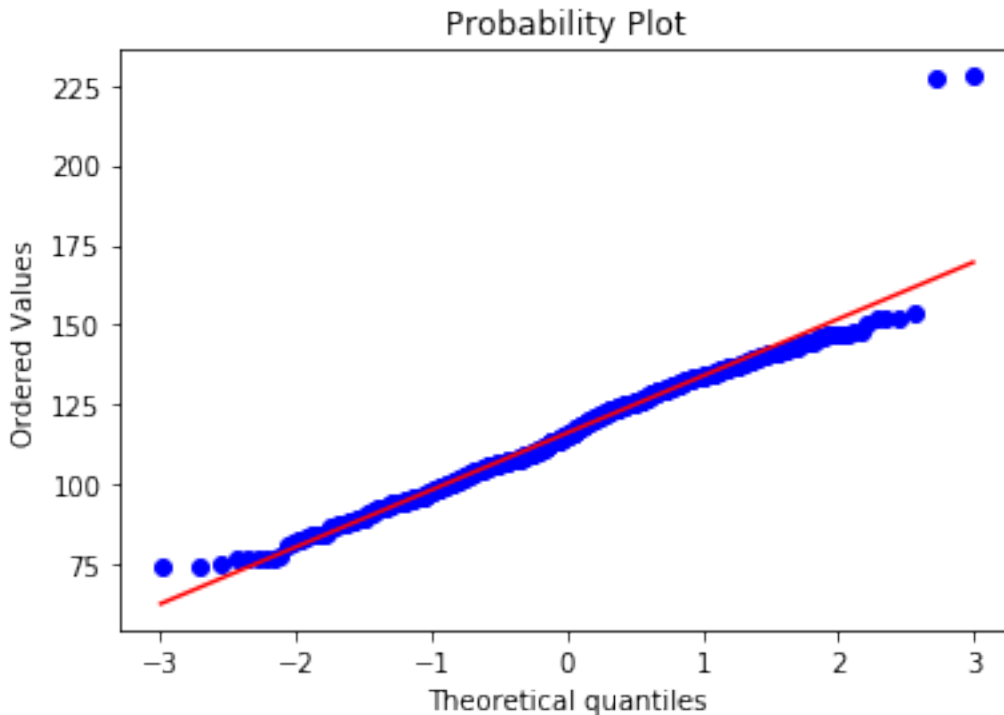
```
[13]: df.boxplot(column="SumbuUtama")
```

```
[13]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac8832548>
```




```
[14]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
→ ")
stats.probplot(df["SumbuUtama"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk")
```

Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk

```
[15]: print("Pengujian normal dengan tes Shapiro-Wilk")
val_list = df["SumbuUtama"].values.tolist()
tset, pval = stats.shapiro(val_list)
print("Pval: ", pval)
if pval > 0.05:
    print("Pval lebih besar dari 0.05, Kesimpulan: Berdistribusi normal")
else:
    print("Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal")
→normal">#Ans
```

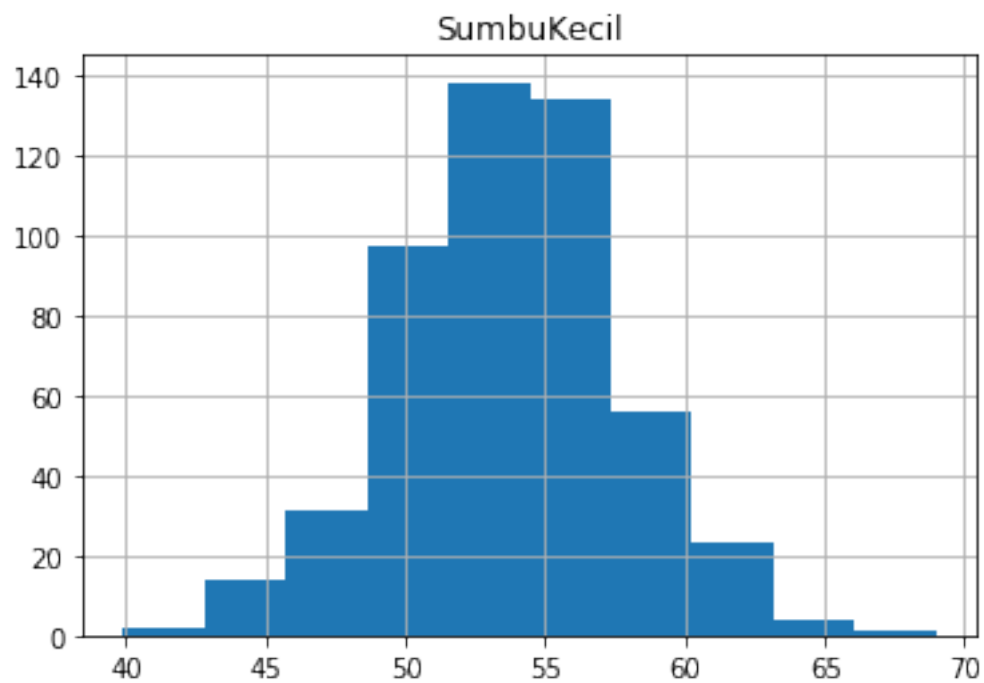
Pengujian normal dengan tes Shapiro-Wilk

Pval: 9.236201213569384e-12

Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal

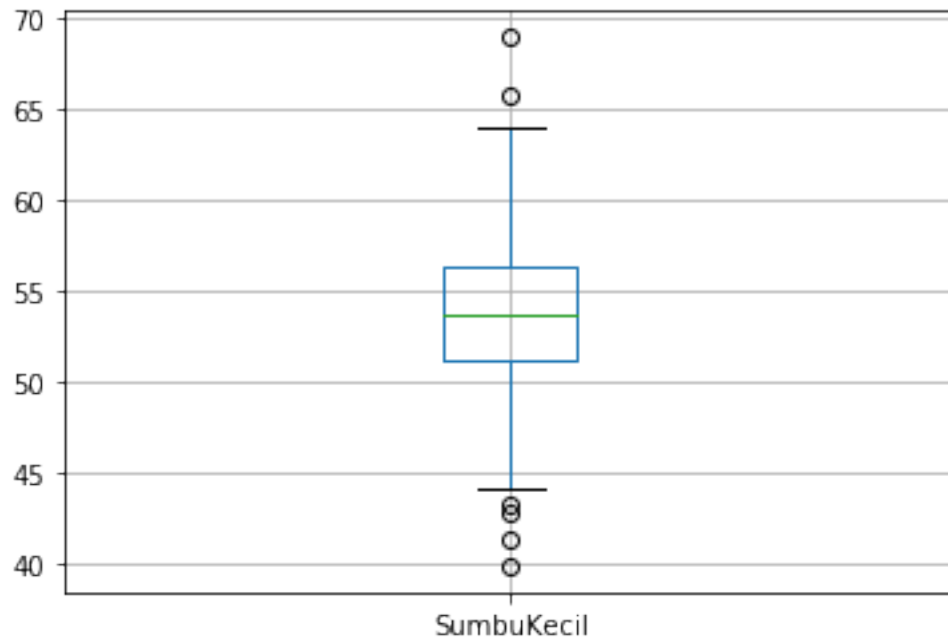
```
[16]: df.hist(column="SumbuKecil")
```

```
[16]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC8916848>]],  
        dtype=object)
```



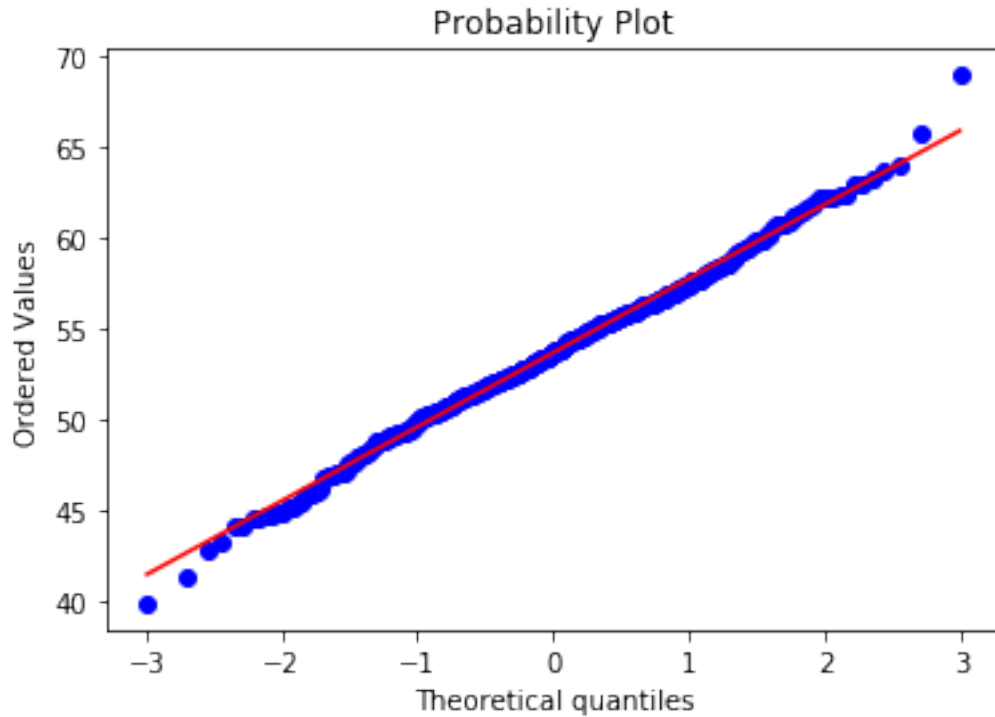
```
[17]: df.boxplot(column="SumbuKecil")
```

```
[17]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac8983a48>
```



```
[18]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
      ↪ ")
stats.probplot(df["SumbuKecil"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk")
```

Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk

```
[19]: print("Pengujian normal dengan tes Shapiro-Wilk")
      val_list = df["SumbuKecil"].values.tolist()
      tset, pval = stats.shapiro(val_list)
      print("Pval: ", pval)
      if pval > 0.05:
          print("Pval lebih besar dari 0.05, Kesimpulan: Berdistribusi normal") #Ans
      else:
          print("Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal")
```

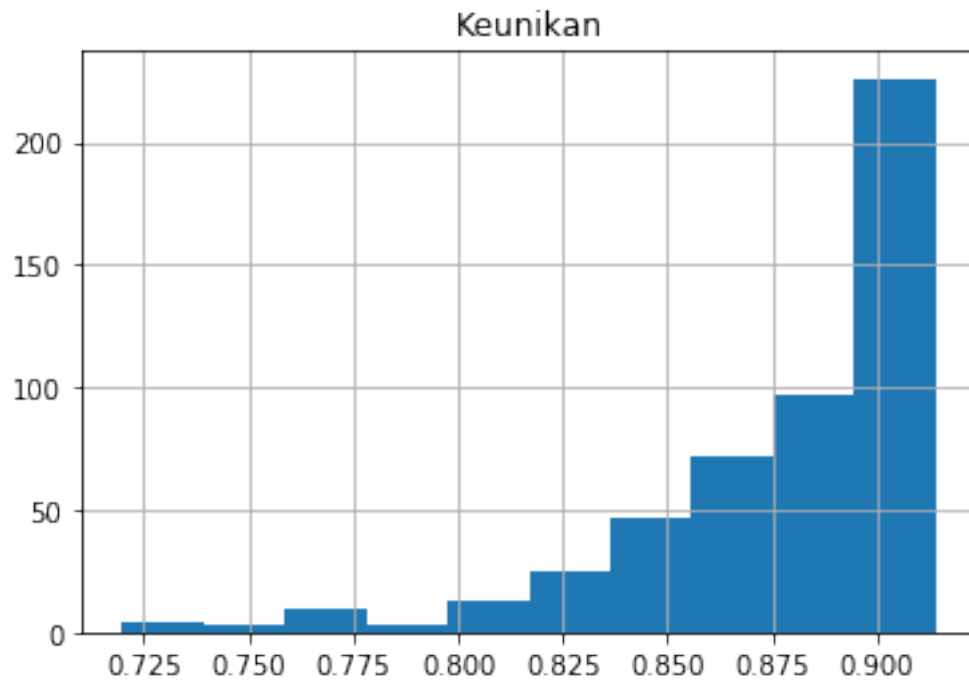
Pengujian normal dengan tes Shapiro-Wilk

Pval: 0.4234558641910553

Pval lebih besar dari 0.05, Kesimpulan: Berdistribusi normal

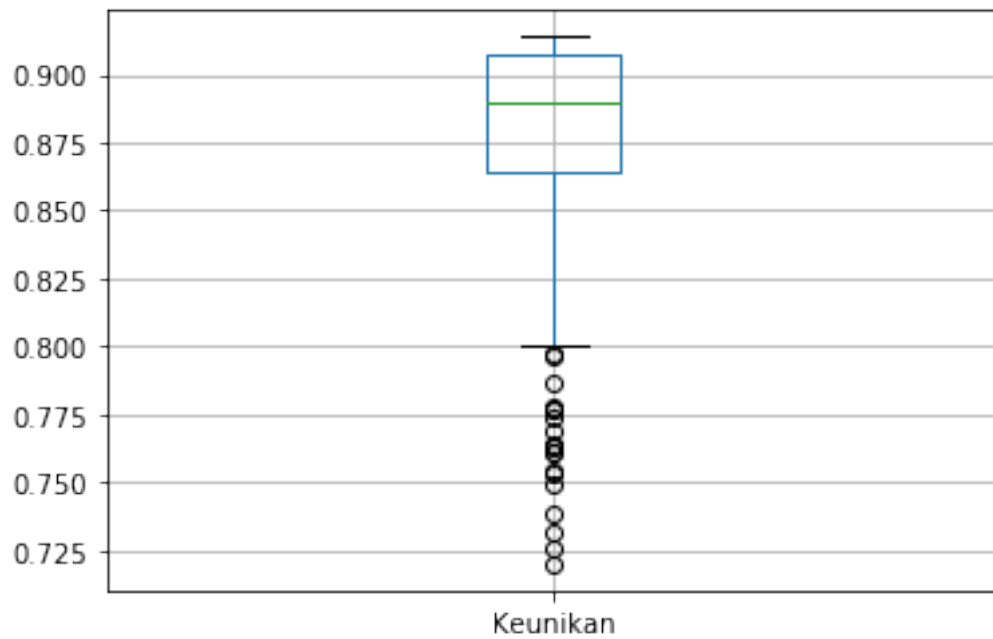
```
[20]: df.hist(column="Keunikan")
```

```
[20]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC8A82588>]],
      dtype=object)
```



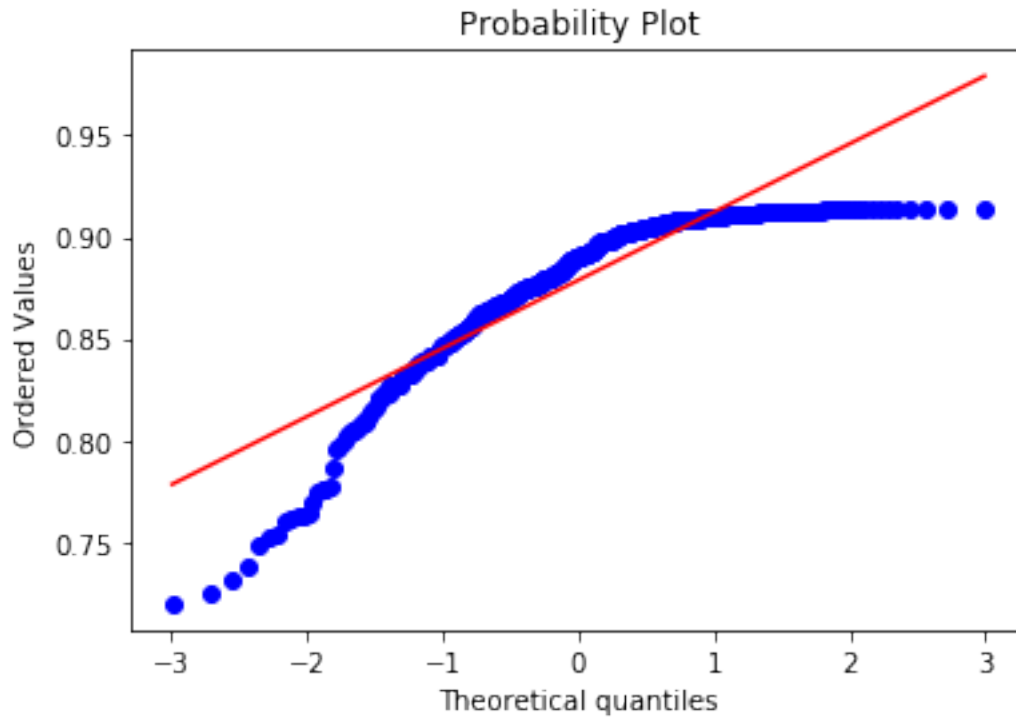
```
[21]: df.boxplot(column="Keunikan")
```

```
[21]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac8b00e48>
```



```
[22]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
→ ")
stats.probplot(df["Keunikan"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Tidak berdistribusi normal")
```

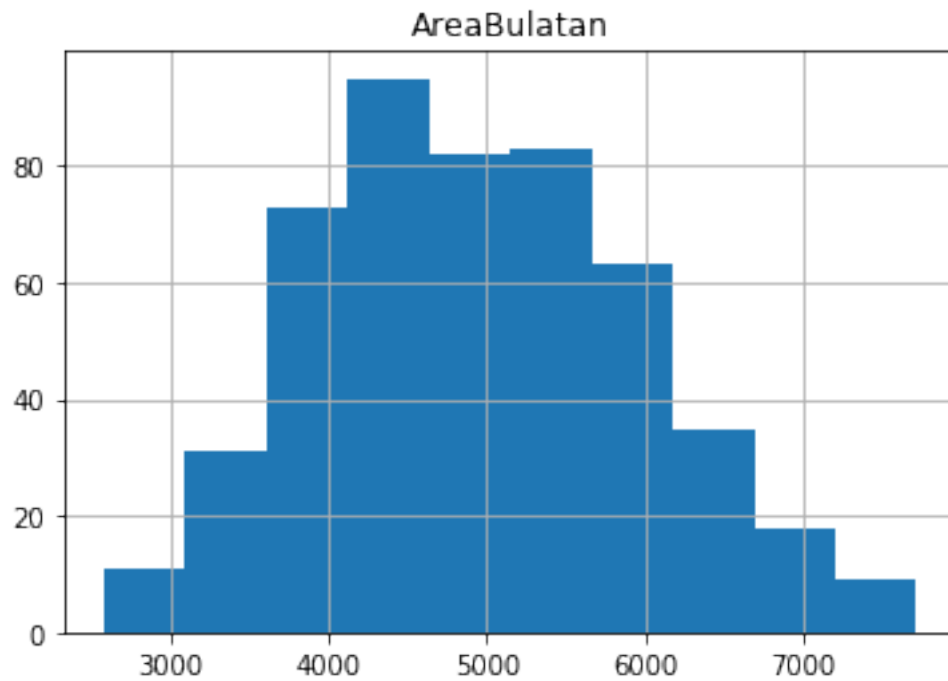
Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Tidak berdistribusi normal

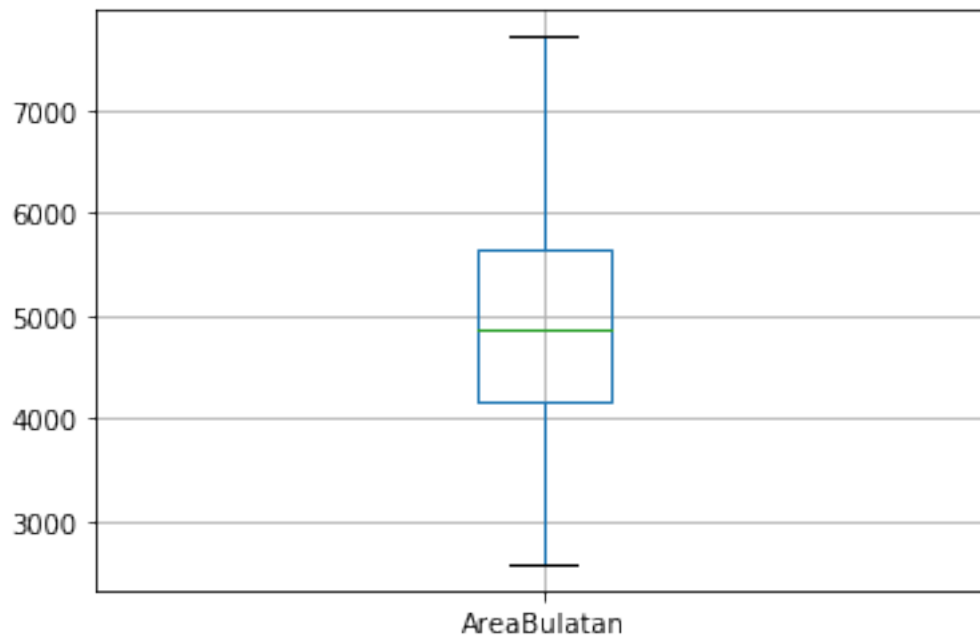
```
[23]: df.hist(column="AreaBulatan")
```

```
[23]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC8A41948>]],
dtype=object)
```



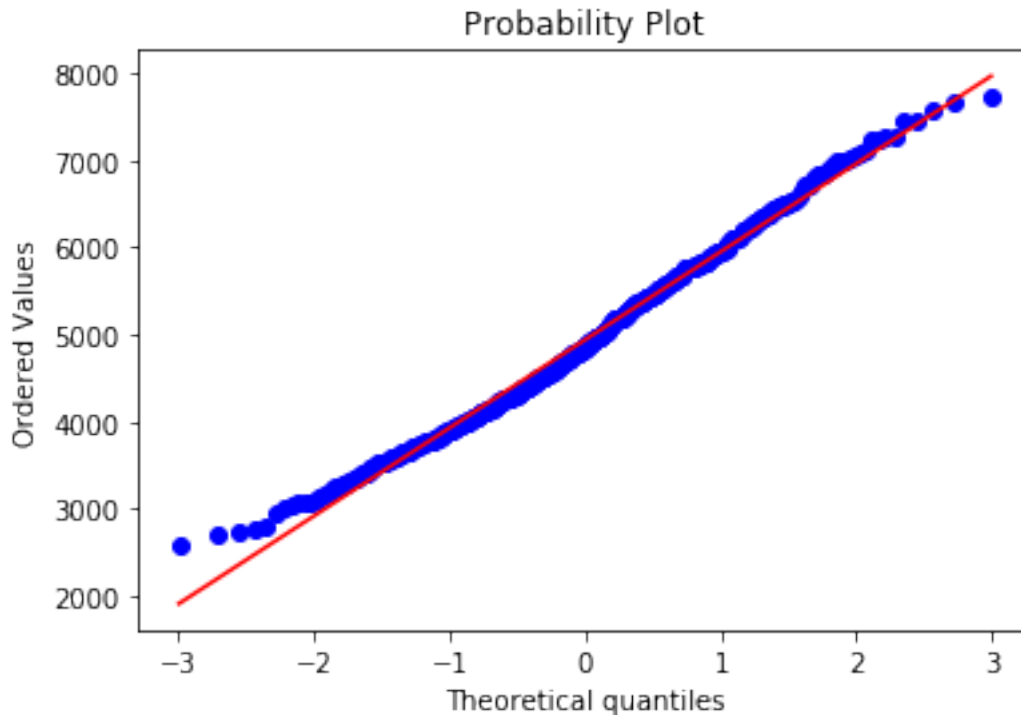
```
[24]: df.boxplot(column="AreaBulatan")
```

```
[24]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac82e9448>
```



```
[25]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
→ ")
stats.probplot(df["AreaBulatan"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk")
```

Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk

```
[26]: print("Pengujian normal dengan tes Shapiro-Wilk")
val_list = df["AreaBulatan"].values.tolist()
tset, pval = stats.shapiro(val_list)
print("Pval: ", pval)
if pval > 0.05:
    print("Pval lebih besar dari 0.05, Kesimpulan: Berdistribusi normal")
else:
    print("Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal")
→ #Ans
```

Pengujian normal dengan tes Shapiro-Wilk

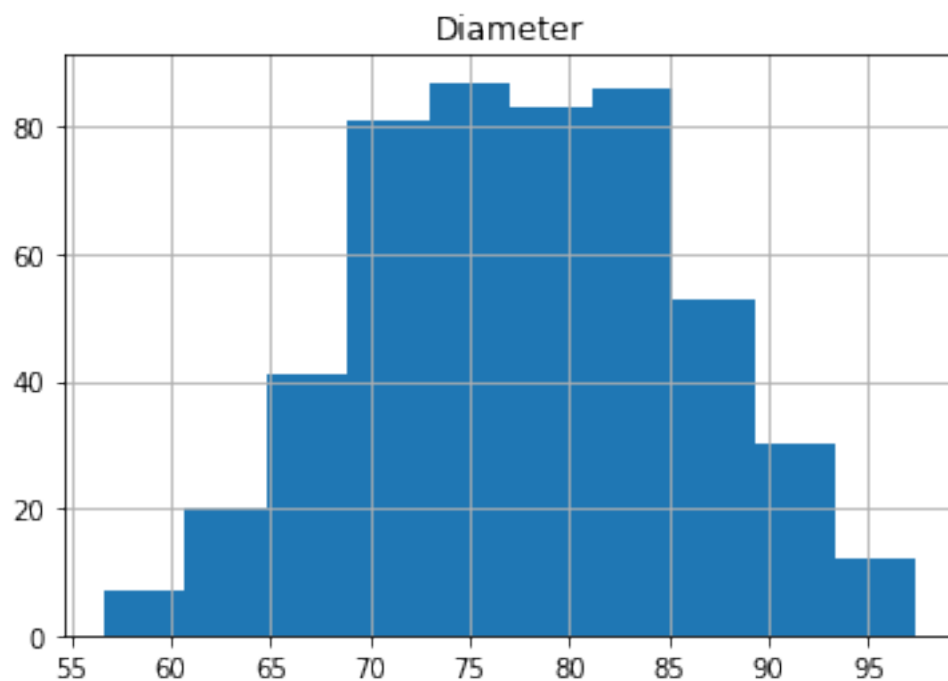
Pval: 0.0024847122840583324

Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal

```
[27]: df.hist(column="Diameter")
```

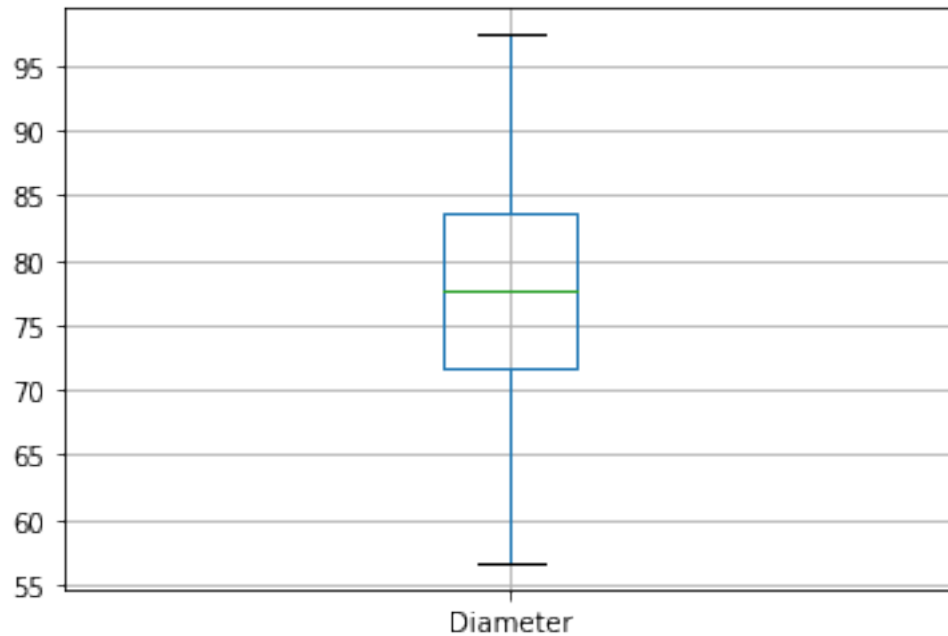


```
[27]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC8BE8188>]],  
        dtype=object)
```



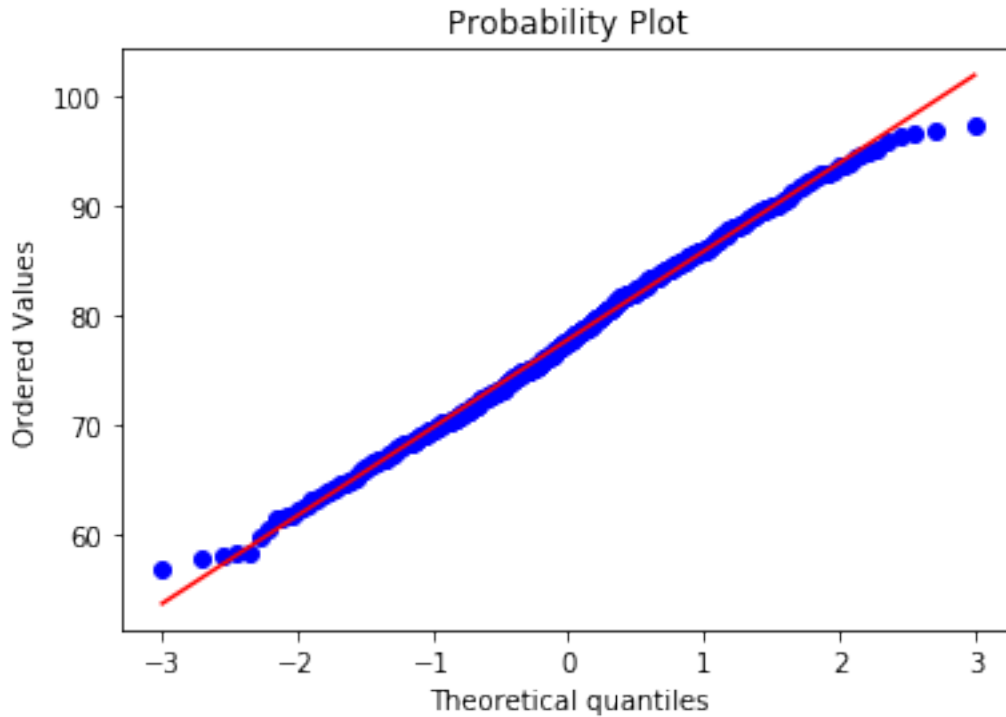
```
[28]: df.boxplot(column="Diameter")
```

```
[28]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac8c74b08>
```



```
[29]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
      ↪ ")
stats.probplot(df["Diameter"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk")
```

Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk

```
[30]: print("Pengujian normal dengan tes Shapiro-Wilk")
      val_list = df["Diameter"].values.tolist()
      tset, pval = stats.shapiro(val_list)
      print("Pval: ", pval)
      if pval > 0.05:
          print("Pval lebih besar dari 0.05, Kesimpulan: Berdistribusi normal") #Ans
      else:
          print("Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal")
```

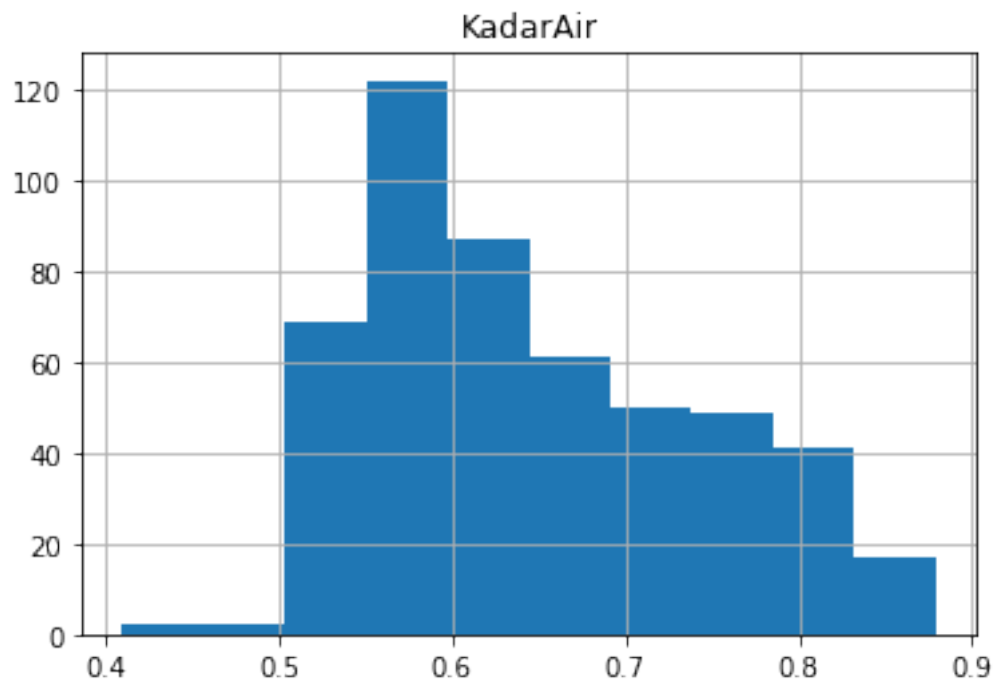
Pengujian normal dengan tes Shapiro-Wilk

Pval: 0.11834503710269928

Pval lebih besar dari 0.05, Kesimpulan: Berdistribusi normal

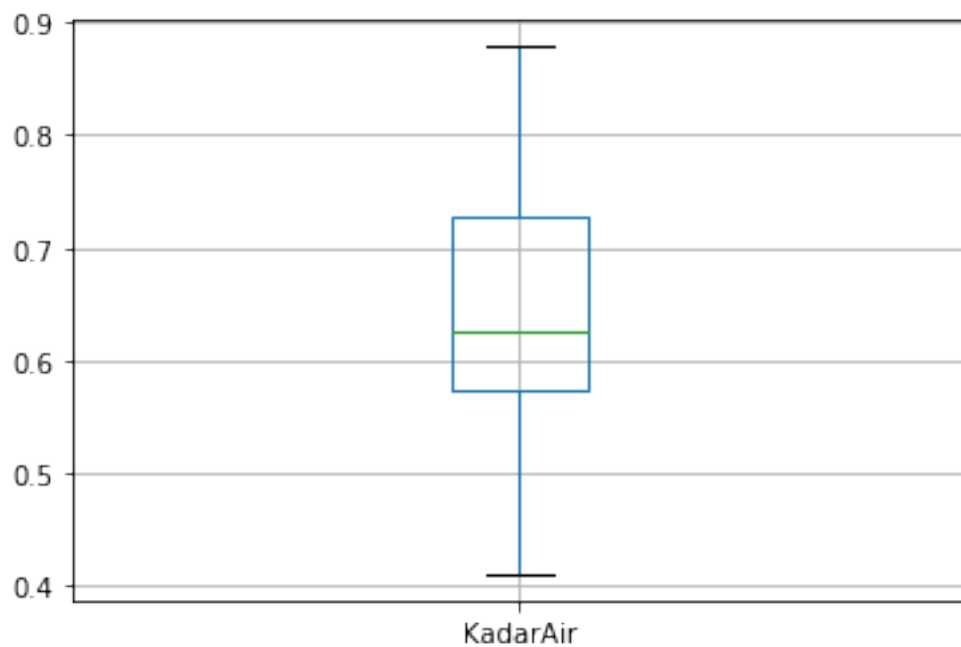
```
[31]: df.hist(column="KadarAir")
```

```
[31]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC8D41688>]],
      dtype=object)
```



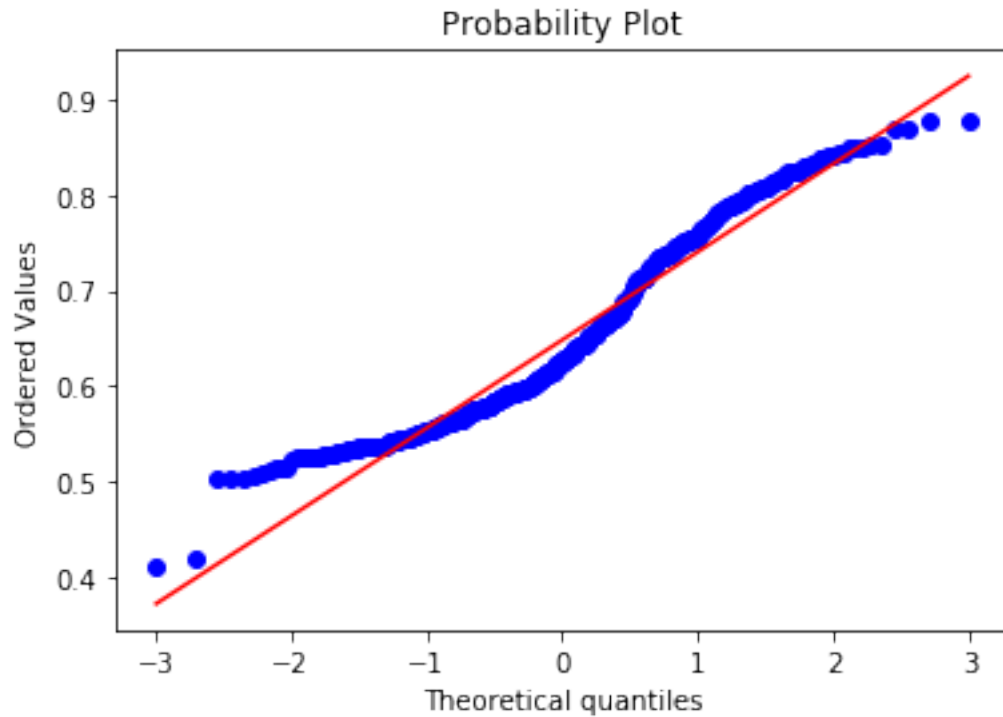
```
[32]: df.boxplot(column="KadarAir")
```

```
[32]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac8dc2348>
```



```
[33]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
→ ")
stats.probplot(df["KadarAir"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Tidak berdistribusi normal")
```

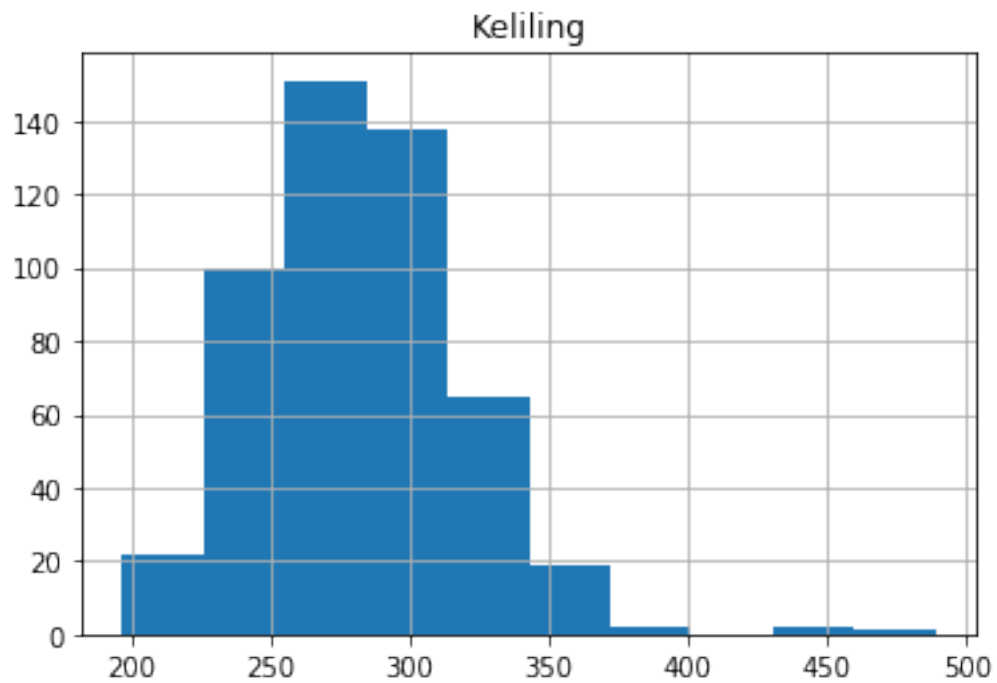
Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Tidak berdistribusi normal

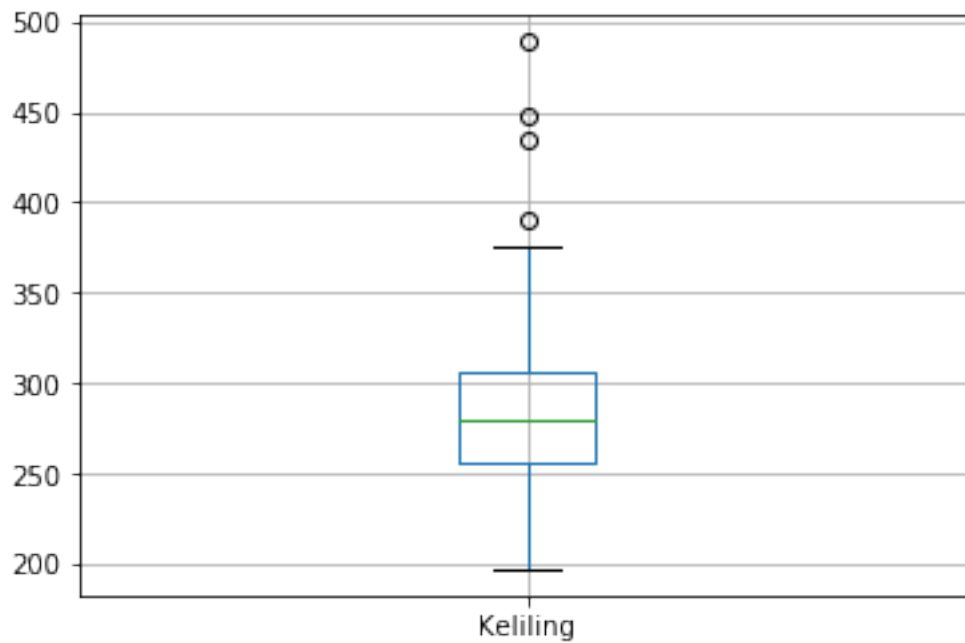
```
[34]: df.hist(column="Keliling")
```

```
[34]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC8EBD308>]],
dtype=object)
```



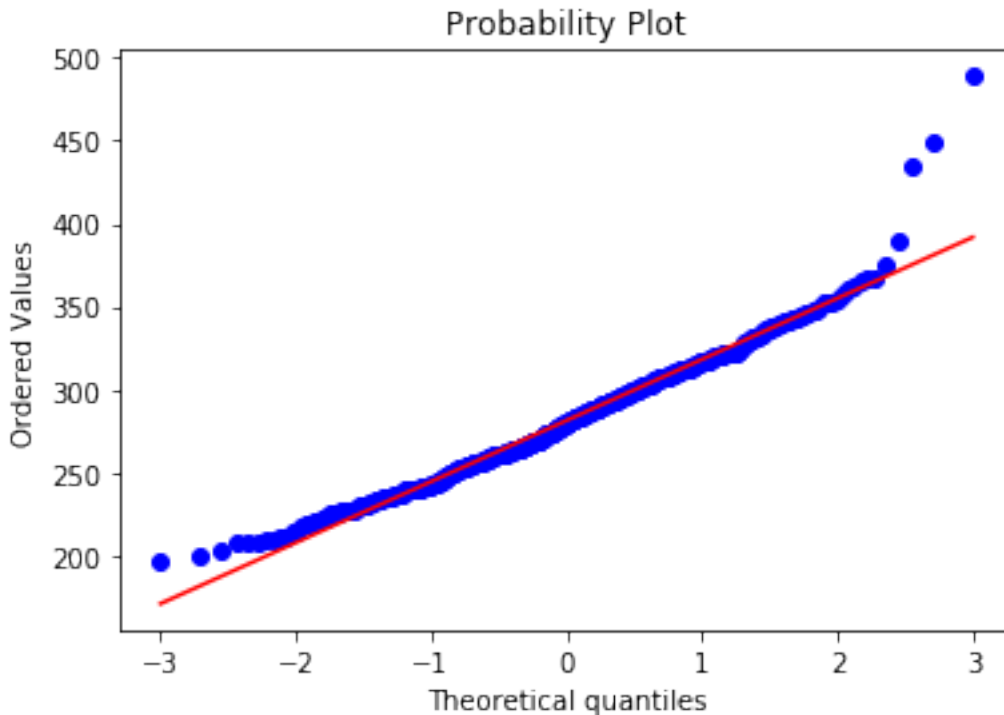
```
[35]: df.boxplot(column="Keliling")
```

```
[35]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac8f41208>
```



```
[36]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
      ↪ ")
stats.probplot(df["Keliling"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk")
```

Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk

```
[37]: print("Pengujian normal dengan tes Shapiro-Wilk")
val_list = df["Keliling"].values.tolist()
tset, pval = stats.shapiro(val_list)
print("Pval: ", pval)
if pval > 0.05:
    print("Pval lebih besar dari 0.05, Kesimpulan: Berdistribusi normal")
else:
    print("Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal")
    ↪ #Ans
```

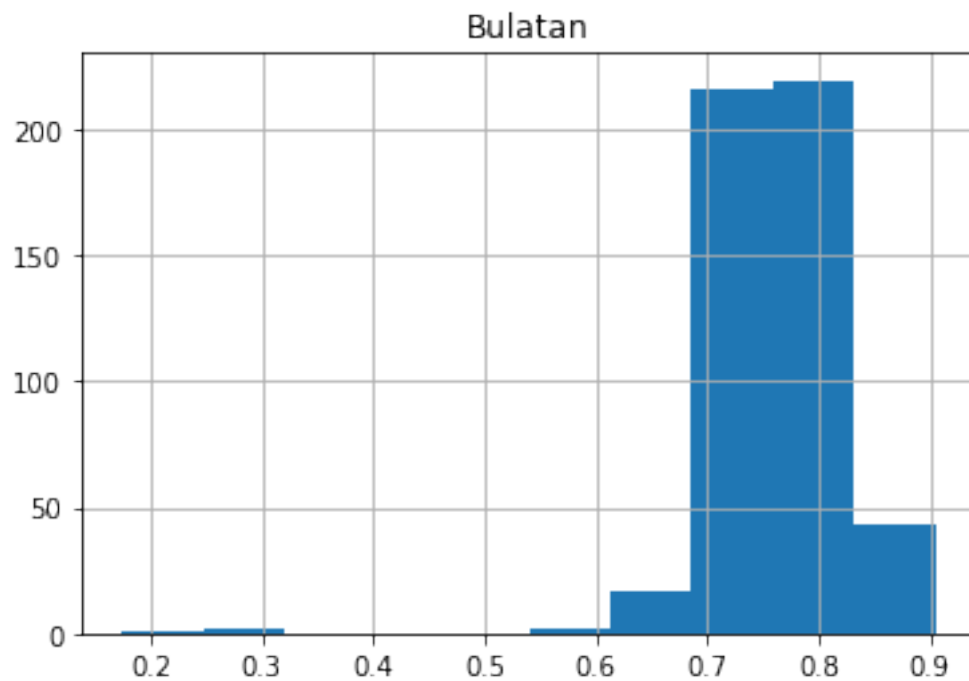
Pengujian normal dengan tes Shapiro-Wilk

Pval: 9.728394090302572e-09

Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal

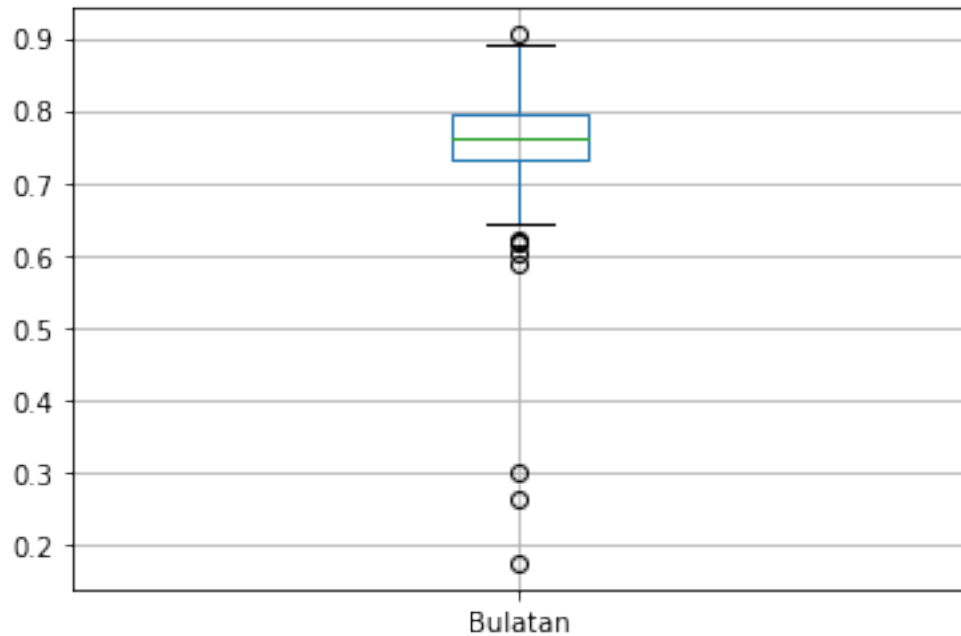
```
[38]: df.hist(column="Bulatan")
```

```
[38]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC8BF64C8>]],  
        dtype=object)
```



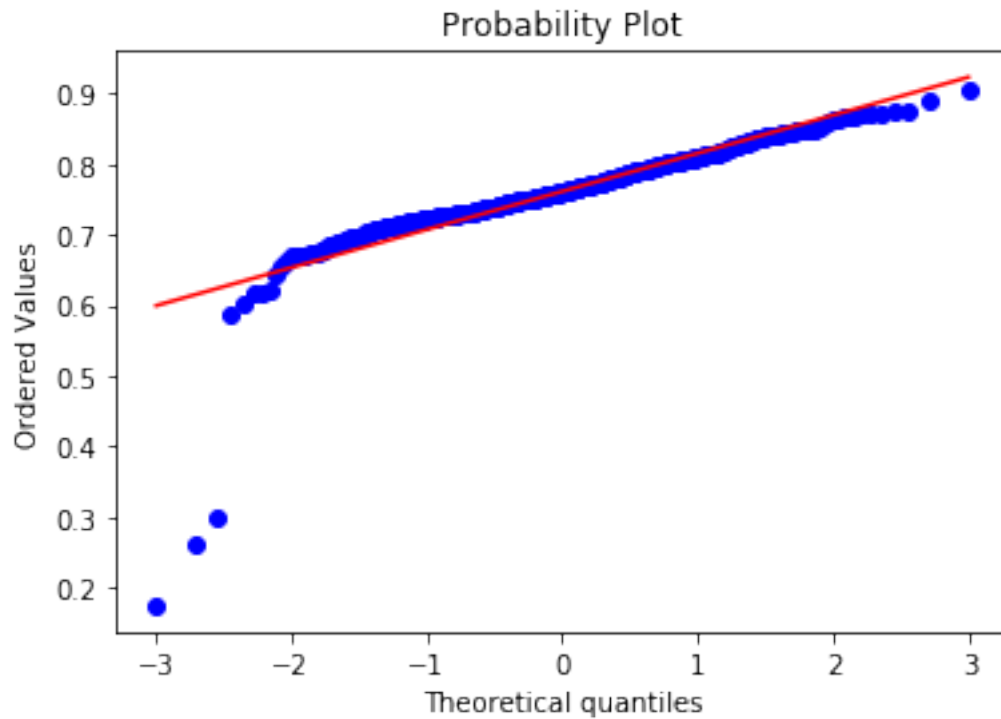
```
[39]: df.boxplot(column="Bulatan")
```

```
[39]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac88e88c8>
```

```
[40]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
      ↪ ")
stats.probplot(df["Bulatan"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk")
```

Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Gagal menolak, perlu dilakukan tes Shapiro-Wilk

```
[41]: print("Pengujian normal dengan tes Shapiro-Wilk")
      val_list = df["Bulatan"].values.tolist()
      tset, pval = stats.shapiro(val_list)
      print("Pval: ", pval)
      if pval > 0.05:
          print("Pval lebih besar dari 0.05, Kesimpulan: Berdistribusi normal")
      else:
          print("Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal") #Ans
```

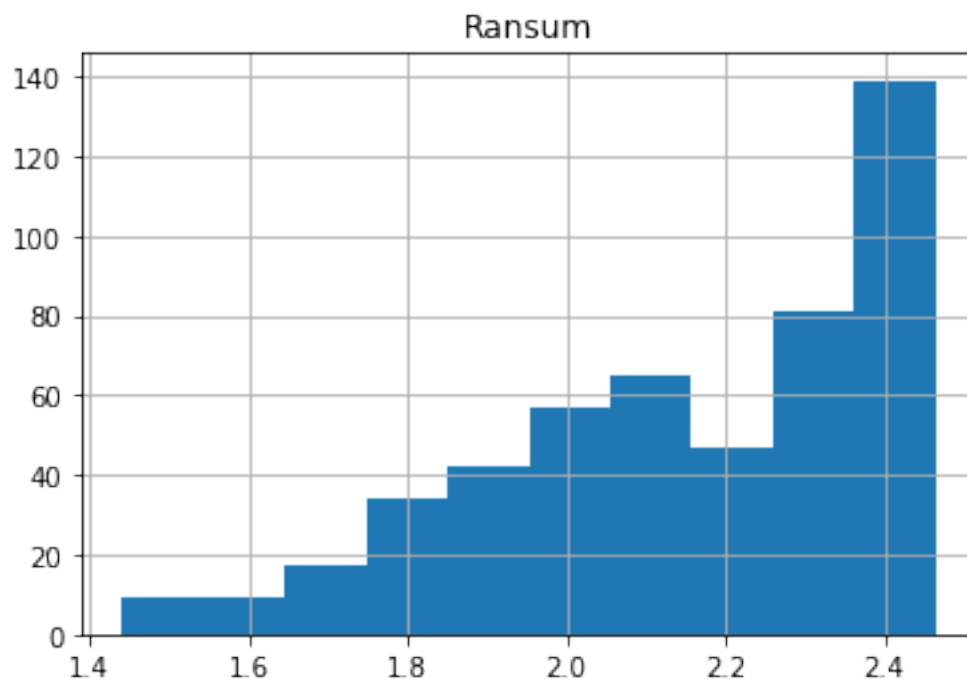
Pengujian normal dengan tes Shapiro-Wilk

Pval: 6.899158691421287e-26

Pval lebih kecil dari 0.05, Kesimpulan: Tidak berdistribusi normal

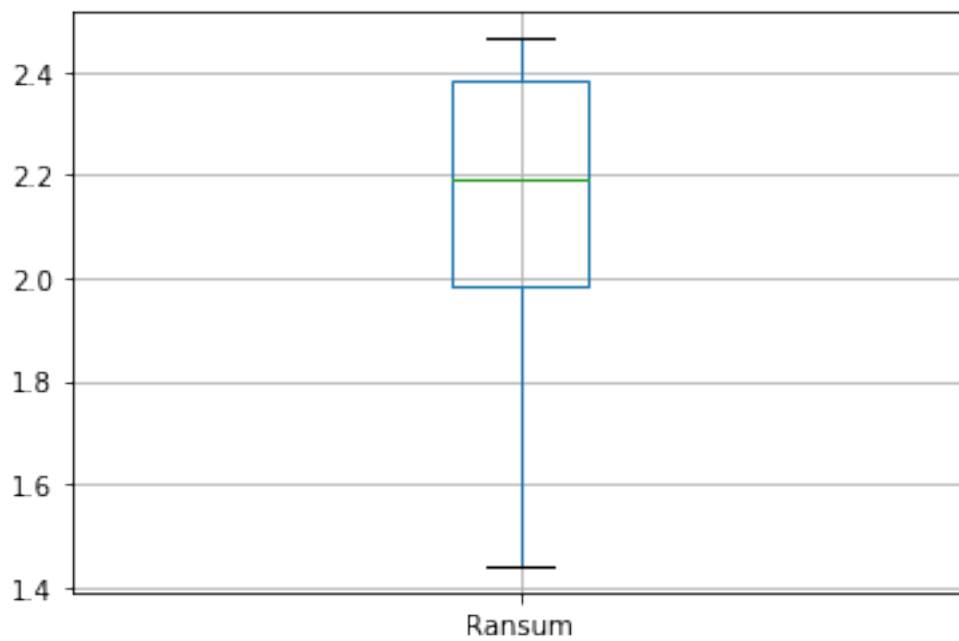
```
[42]: df.hist(column="Ransum")
```

```
[42]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015AC8FCE888>]],
      dtype=object)
```



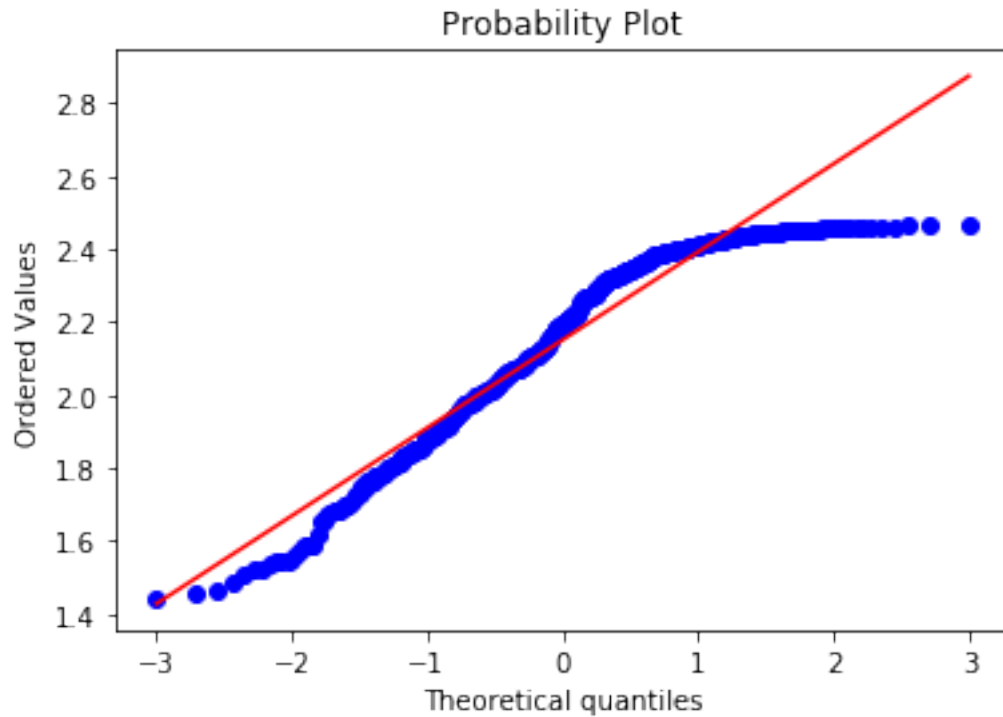
```
[43]: df.boxplot(column="Ransum")
```

```
[43]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac9064848>
```



```
[44]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
→ ")
stats.probplot(df["Ransum"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Tidak berdistribusi normal")
```

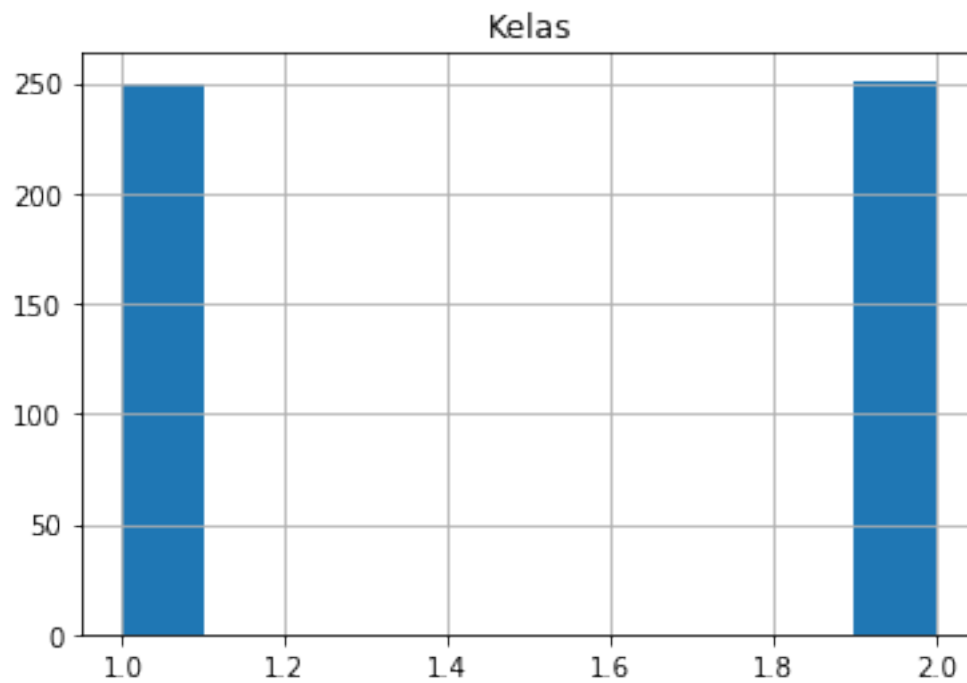
Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



Kesimpulan: Tidak berdistribusi normal

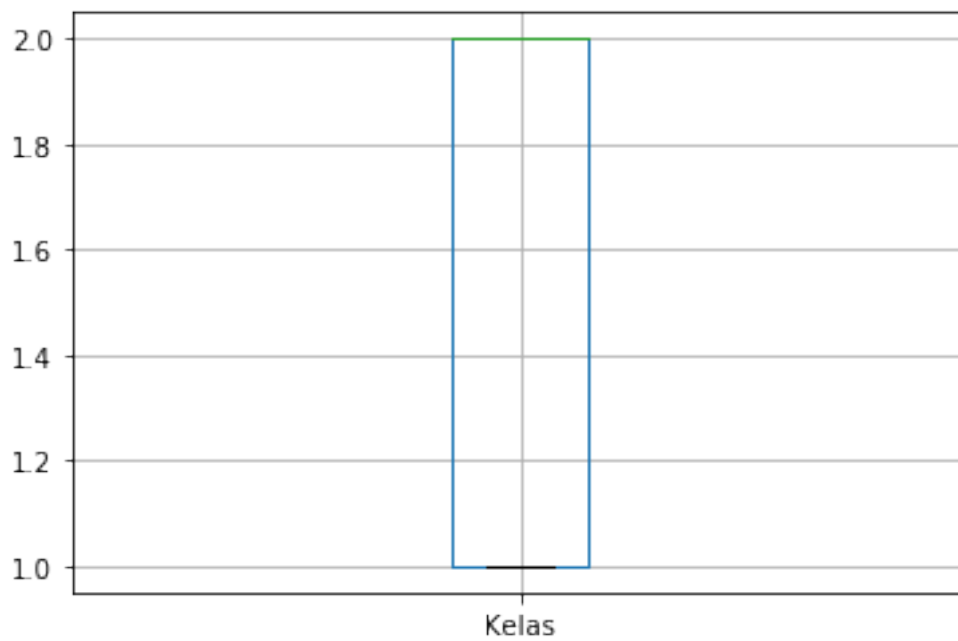
```
[45]: df.hist(column="Kelas")
```

```
[45]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000015ACA0D75C8>]],
dtype=object)
```



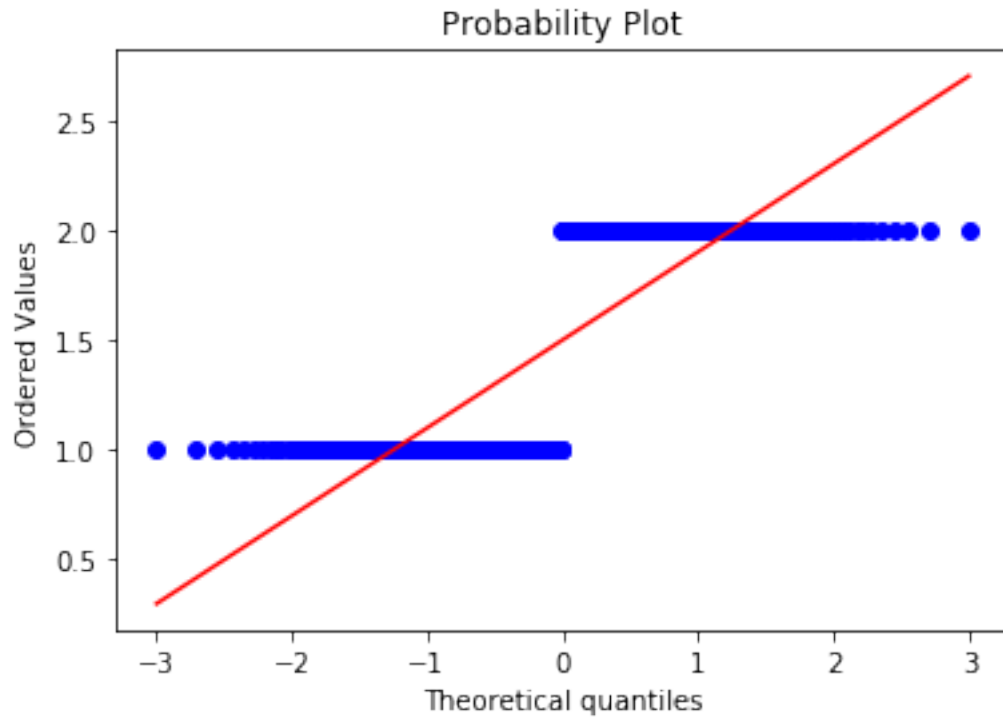
```
[46]: df.boxplot(column="Kelas")
```

```
[46]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca195d08>
```



```
[47]: print("Pengujian distribusi normal dengan normality test Quartile-Quartile plot:
      ↪ ")
stats.probplot(df["Kelas"], dist="norm", plot=pylab)
pylab.show()
print("Kesimpulan: Tidak berdistribusi normal")
```

Pengujian distribusi normal dengan normality test Quartile-Quartile plot:



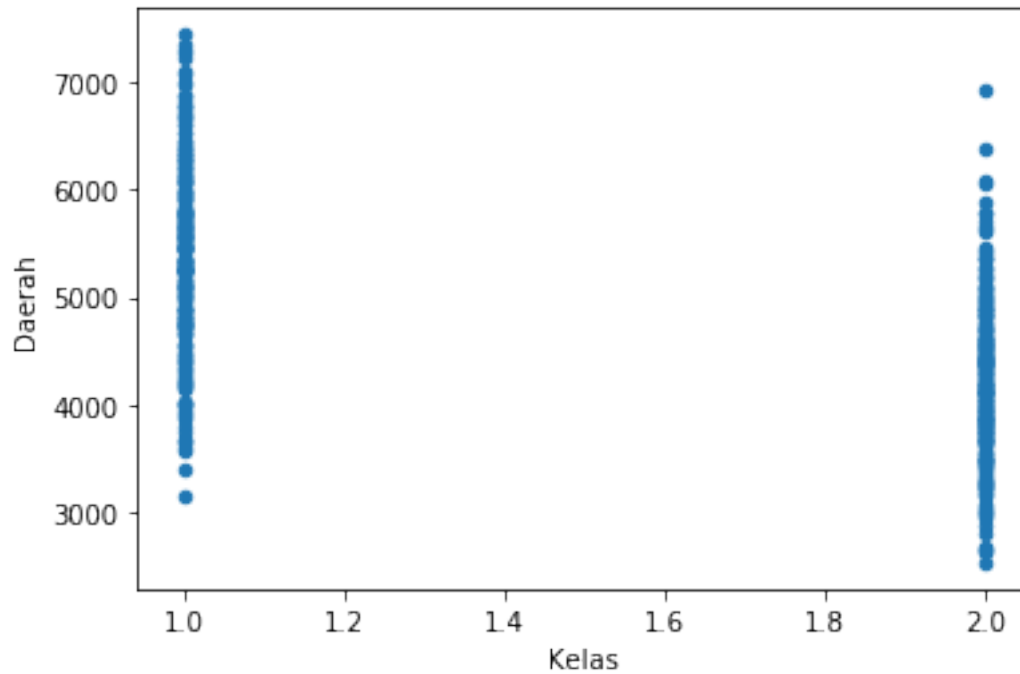
Kesimpulan: Tidak berdistribusi normal

```
[48]: #Korelasi
      #Target adalah kolom "Kelas", yang memiliki nilai 1 atau 2
      print("Correlation")
```

Correlation

```
[49]: #Kelas-Daerah
      df.plot.scatter("Kelas", "Daerah")
```

```
[49]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca1c9508>
```



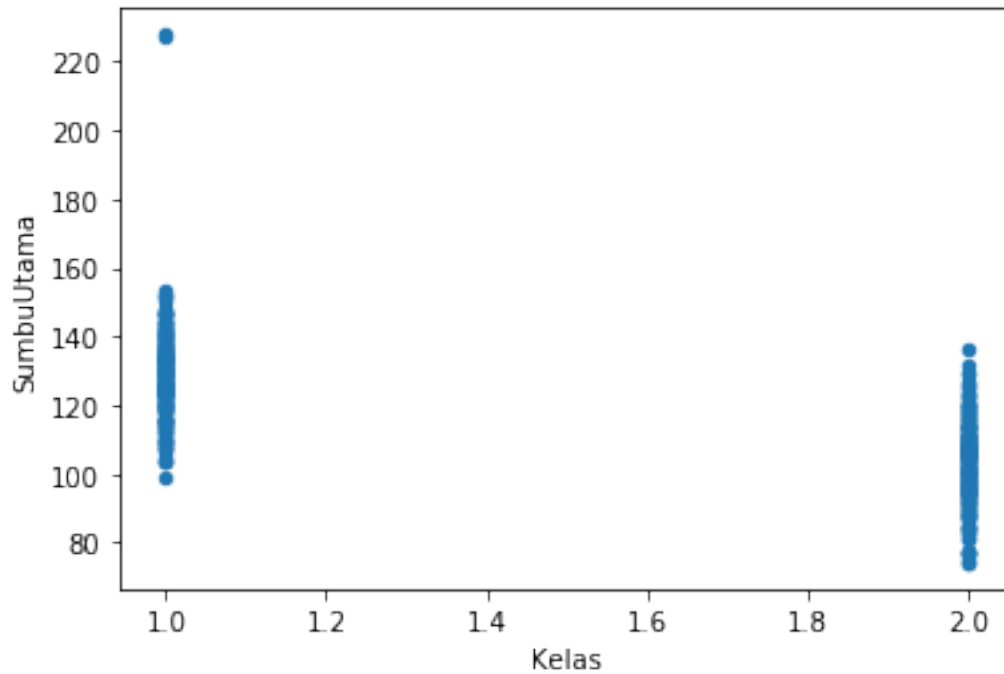
```
[50]: print("Korelasi negatif kuat (0.6-0.79)")
      print("Nilai korelasi: ")
      df["Kelas"].corr(df["Daerah"])
```

Korelasi negatif kuat (0.6-0.79)
 Nilai korelasi:

```
[50]: -0.6027466517416654
```

```
[51]: #Kelas-SumbuUtama
      df.plot.scatter("Kelas", "SumbuUtama")
```

```
[51]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca2c2908>
```



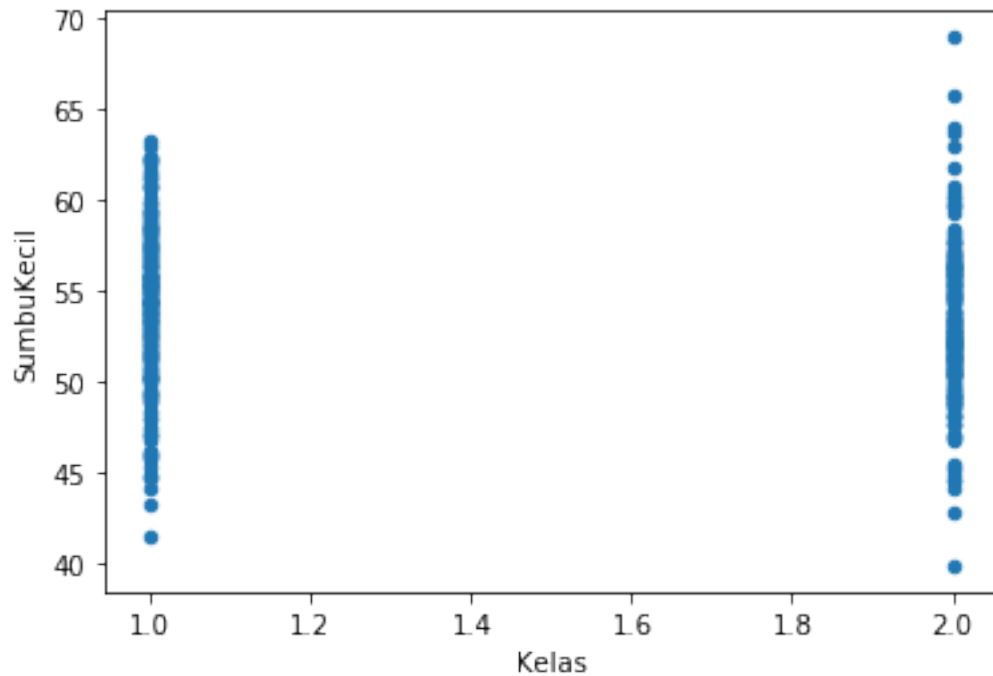
```
[52]: print("Korelasi negatif kuat (0.6-0.79)")
      print("Nilai korelasi: ")
      df["Kelas"].corr(df["SumbuUtama"])
```

Korelasi negatif kuat (0.6-0.79)
 Nilai korelasi:

```
[52]: -0.7130906104204592
```

```
[53]: #Kelas-SumbuKecil
      df.plot.scatter("Kelas", "SumbuKecil")
```

```
[53]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca356b08>
```

```
[54]: print("Korelasi negatif sangat lemah (0.0-0.19)")
      print("Nilai korelasi: ")
      df["Kelas"].corr(df["SumbuKecil"])
```

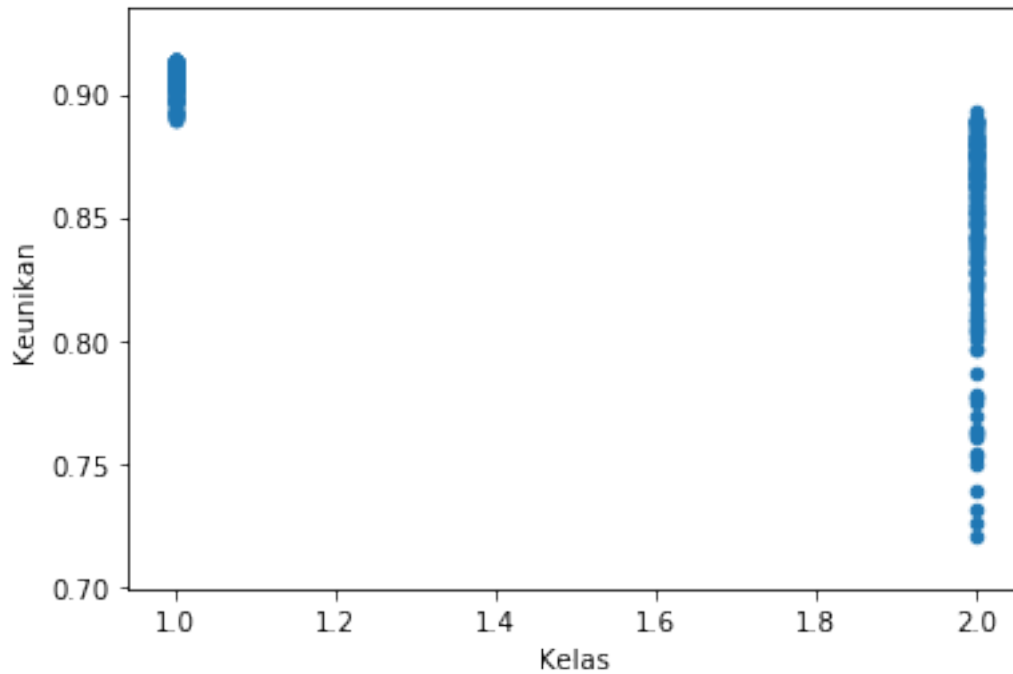
Korelasi negatif sangat lemah (0.0-0.19)

Nilai korelasi:

```
[54]: -0.15297517335535024
```

```
[55]: #Kelas-Keunikan
      df.plot.scatter("Kelas", "Keunikan")
```

```
[55]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca3b8748>
```



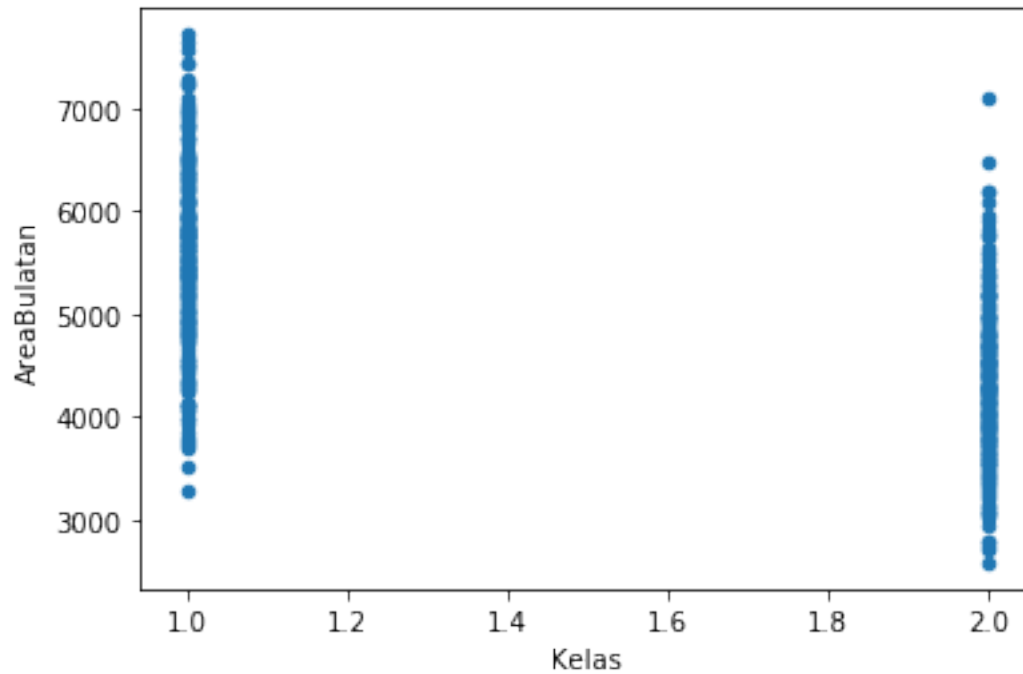
```
[56]: print("Korelasi negatif kuat (0.6-0.79)")
      print("Nilai korelasi: ")
      df["Kelas"].corr(df["Keunikan"])
```

Korelasi negatif kuat (0.6-0.79)
Nilai korelasi:

```
[56]: -0.7304563686511927
```

```
[57]: #Kelas-AreaBulatan
      df.plot.scatter("Kelas", "AreaBulatan")
```

```
[57]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca3f9648>
```



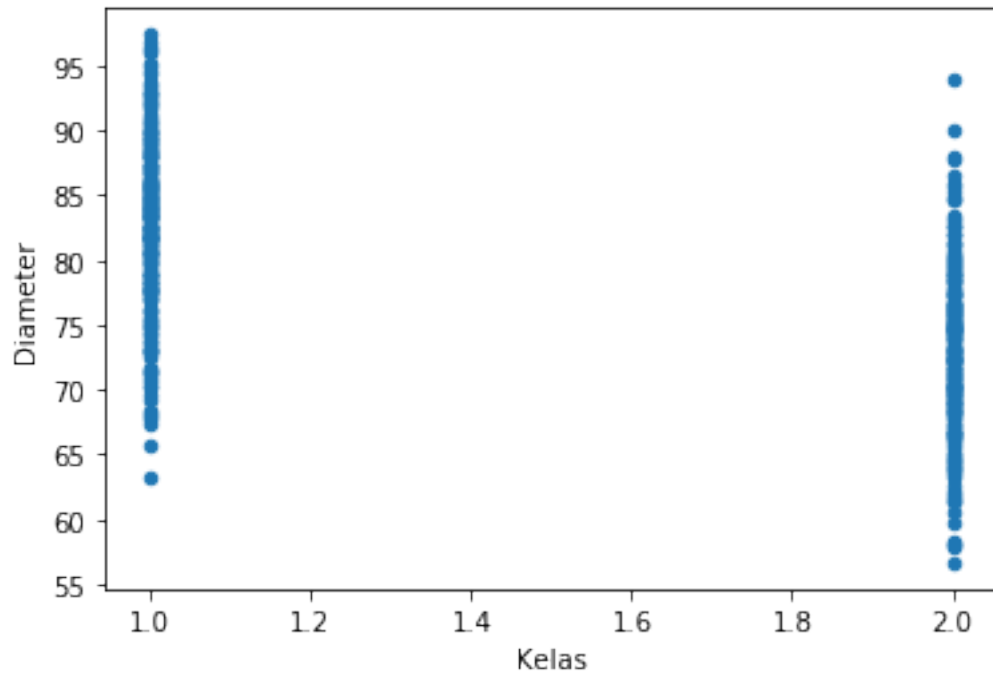
```
[58]: print("Korelasi negatif kuat (0.6-0.79)")
      print("Nilai korelasi: ")
      df["Kelas"].corr(df["AreaBulatan"])
```

Korelasi negatif kuat (0.6-0.79)
 Nilai korelasi:

```
[58]: -0.6073125434153747
```

```
[59]: #Kelas-Diameter
      df.plot.scatter("Kelas", "Diameter")
```

```
[59]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca0eafc8>
```



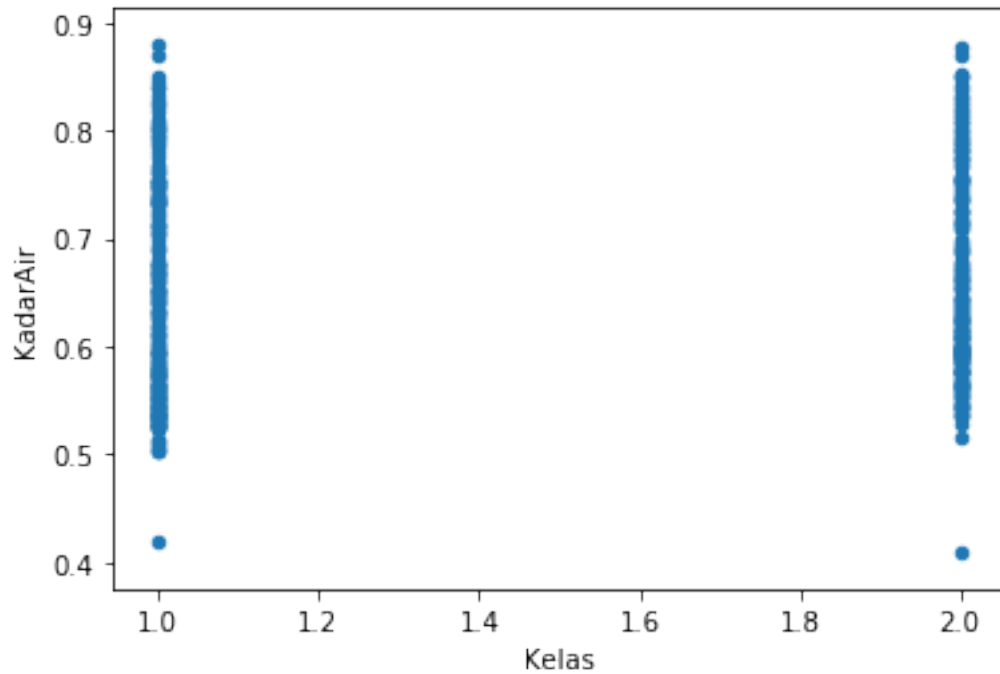
```
[60]: print("Korelasi negatif kuat (0.6-0.79)")  
      print("Nilai korelasi: ")  
      df["Kelas"].corr(df["Diameter"])
```

```
Korelasi negatif kuat (0.6-0.79)  
Nilai korelasi:
```

```
[60]: -0.6025356896618811
```

```
[61]: #Kelas-KadarAir  
      df.plot.scatter("Kelas", "KadarAir")
```

```
[61]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac8fab248>
```



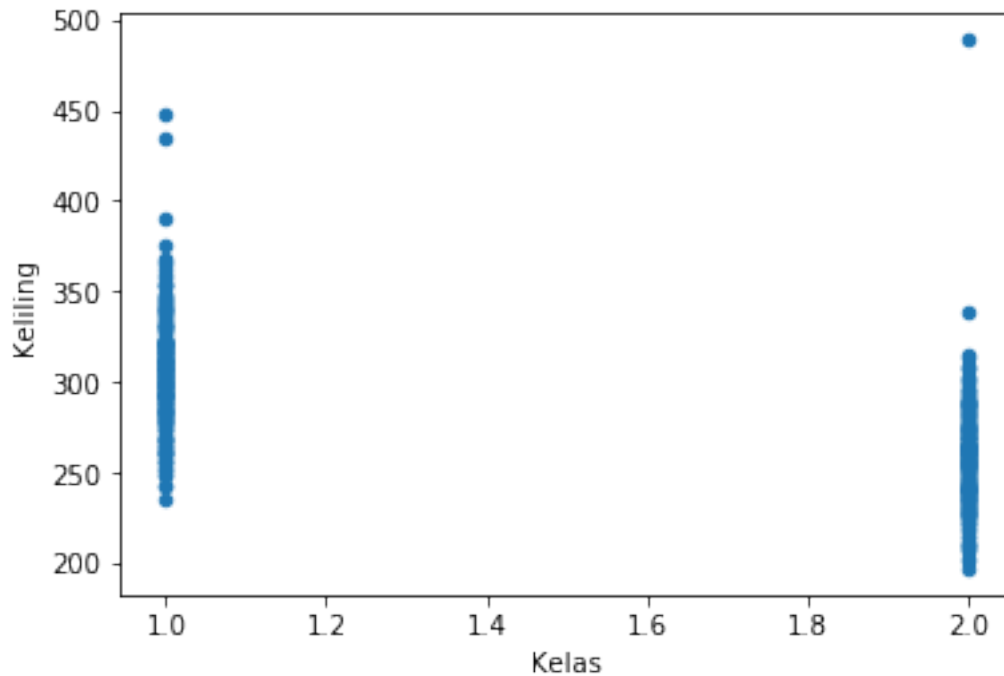
```
[62]: print("Korelasi positif sangat lemah (0.0-0.19)")
      print("Nilai korelasi: ")
      df["Kelas"].corr(df["KadarAir"])
```

```
Korelasi positif sangat lemah (0.0-0.19)
Nilai korelasi:
```

```
[62]: 0.13434422605727644
```

```
[63]: #Kelas-Keliling
      df.plot.scatter("Kelas", "Keliling")
```

```
[63]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac8d2ea48>
```



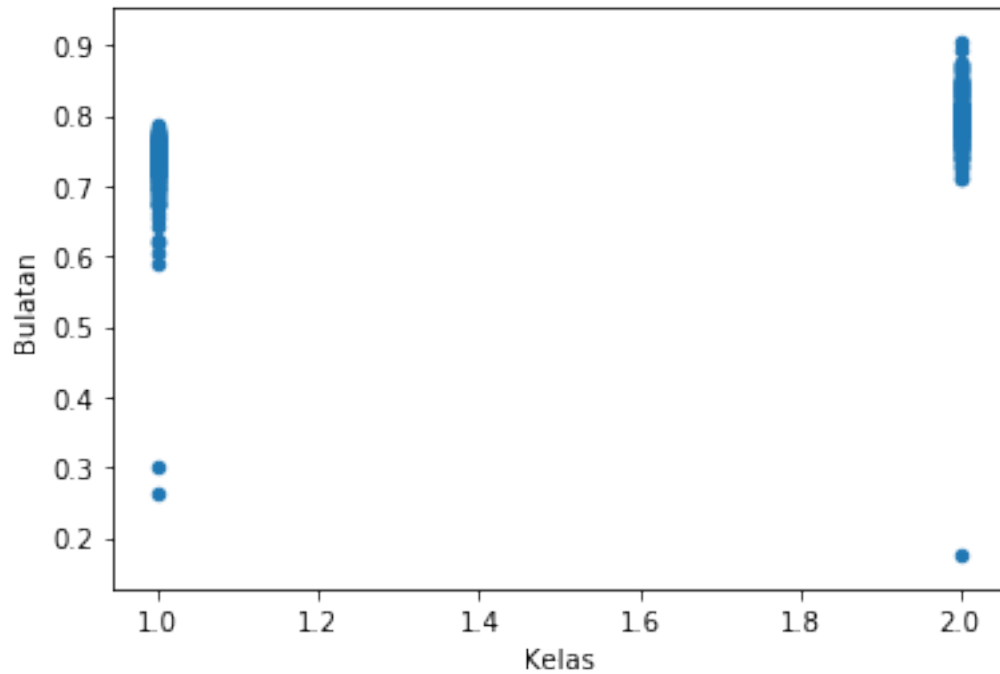
```
[64]: print("Korelasi negatif kuat (0.6-0.79)")
      print("Nilai korelasi: ")
      df["Kelas"].corr(df["Keliling"])
```

Korelasi negatif kuat (0.6-0.79)
 Nilai korelasi:

```
[64]: -0.6348607454756852
```

```
[65]: #Kelas-Bulatan
      df.plot.scatter("Kelas", "Bulatan")
```

```
[65]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca2111c8>
```



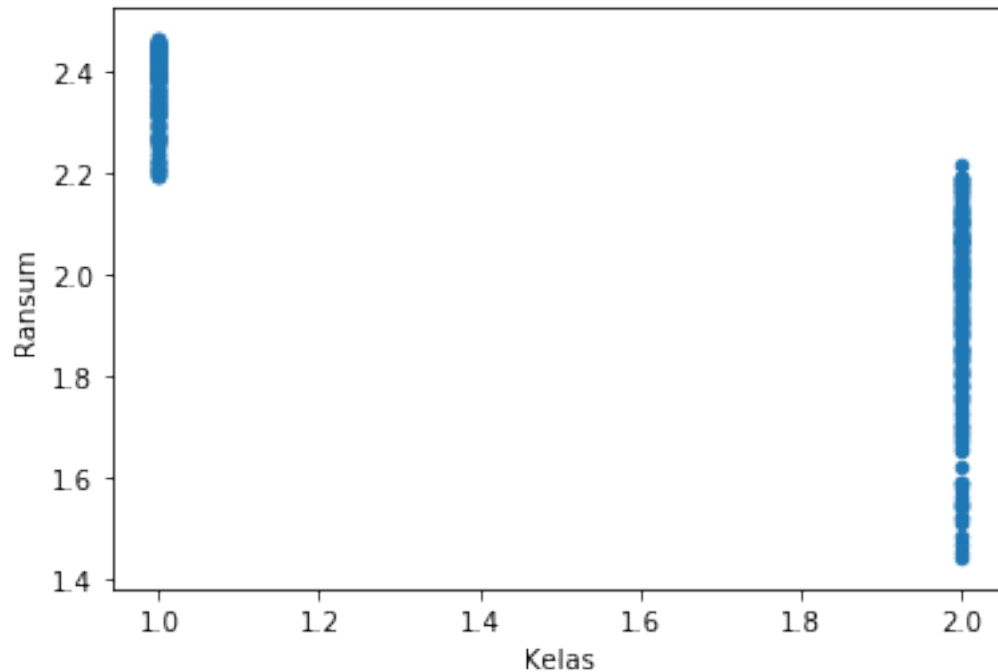
```
[66]: print("Korelasi positif sedang (0.4-0.59)")
      print("Nilai korelasi: ")
      df["Kelas"].corr(df["Bulatan"])
```

Korelasi positif sedang (0.4-0.59)
Nilai korelasi:

```
[66]: 0.5450045317240073
```

```
[67]: #Kelas-Ransum
      df.plot.scatter("Kelas", "Ransum")
```

```
[67]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca4730c8>
```



```
[68]: print("Korelasi negatif sangat kuat (0.8-1.0)")
print("Nilai korelasi: ")
df["Kelas"].corr(df["Ransum"])
```

Korelasi negatif sangat kuat (0.8-1.0)
 Nilai korelasi:

```
[68]: -0.8399038681287486
```

```
[69]: print("Tes Hipotesis 1 Sampel")
print("4.a: Nilai rata-rata Daerah di atas 4700?")
print("h0: rata-rata = 4700")
print("h1: rata-rata > 4700")
print("alpha = 0.05")
print("Tes yang digunakan: One Sample One-Tailed Normal test")
print("Daerah kritis: Z > 1.645")
val_list = df["Daerah"].values.tolist()
tset, pval = stats.ttest_1samp(val_list, 4700)
print("Nilai P: ", pval)
# tes yang diperlukan one-tailed, sementara fungsi menghitung two-ended,
# sehingga P daerah kritis perlu 2x lipat
# note: keberadaan tail kedua dinilai tidak memengaruhi hasil
if pval < 0.10:
    print("Pvalue lebih kecil dari alpha, Hipotesis null tidak diterima,
    sehingga rata-rata diatas 4700")
```



```

else:
    print("Pvalue lebih besar dari alpha, Hipotesis null diterima, sehingga rata-rata 4700")
df.boxplot(column="Daerah")

```

Tes Hipotesis 1 Sampel

4.a: Nilai rata-rata Daerah di atas 4700?

h_0 : rata-rata = 4700

h_1 : rata-rata > 4700

$\alpha = 0.05$

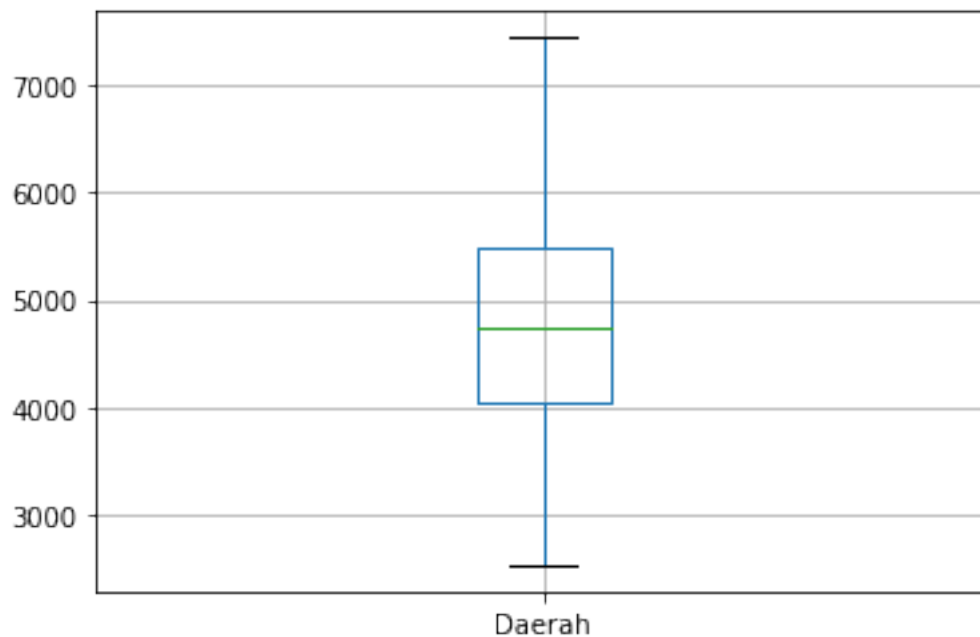
Tes yang digunakan: One Sample One-Tailed Normal test

Daerah kritis: $Z > 1.645$

Nilai P: 0.022138370895226352

Pvalue lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga rata-rata diatas 4700

[69]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca4d3ec8>



```

[70]: print("4.b: Nilai Rata-rata Sumbu Utama tidak sama dengan 116?")
      print("h0: rata-rata = 116")
      print("h1: rata-rata != 116")
      print("alpha = 0.05")
      print("Tes yang digunakan: One Sample Two-Tailed Normal test")
      print("Daerah kritis: Z > 1.96 atau Z < -1.96")
      val_list = df["SumbuUtama"].values.tolist()
      tset, pval = stats.ttest_1samp(val_list, 116)

```

```

print("Nilai P: ", pval)
if pval < 0.05:
    print("Pvalue lebih kecil dari alpha, Hipotesis null tidak diterima,
    ↳ sehingga rata-rata bukan 116")
else:
    print("Pvalue lebih besar dari alpha, Hipotesis null diterima, sehingga
    ↳ rata-rata 116")
df.boxplot(column="SumbuUtama")

```

4.b: Nilai Rata-rata Sumbu Utama tidak sama dengan 116?

H_0 : rata-rata = 116

H_1 : rata-rata \neq 116

α = 0.05

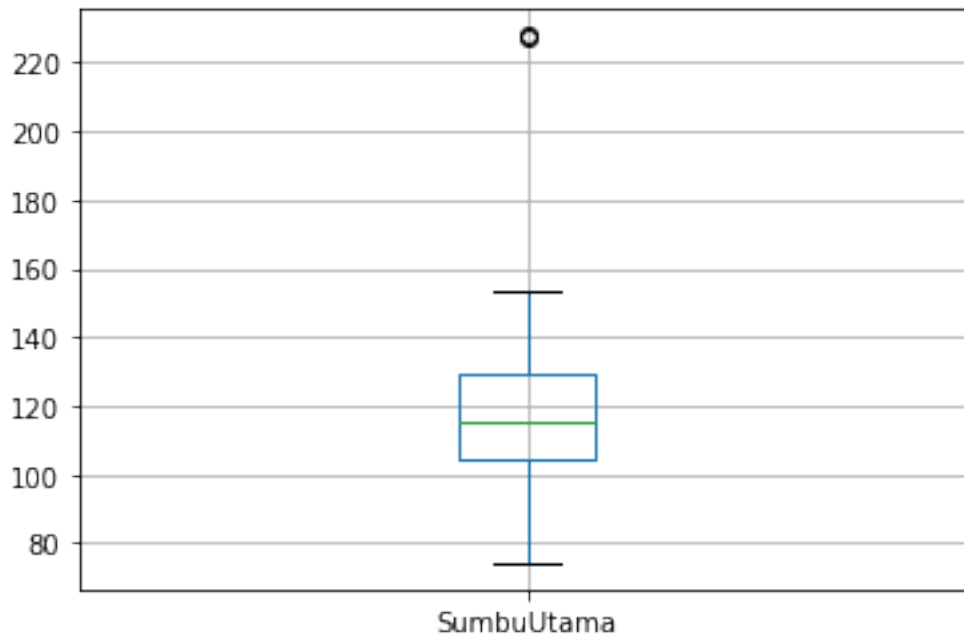
Tes yang digunakan: One Sample Two-Tailed Normal test

Daerah kritis: $Z > 1.96$ atau $Z < -1.96$

Nilai P: 0.955963699941099

Pvalue lebih besar dari alpha, Hipotesis null diterima, sehingga rata-rata 116

[70]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca5184c8>



```

[71]: print("4c. Nilai Rata-rata 20 baris pertama kolom Sumbu Kecil bukan 50?")
print("h0: rata-rata = 50")
print("h1: rata-rata  $\neq$  50")
print("alpha = 0.05")
print("Tes yang digunakan: One Sample Two-Tailed Normal test")
print("Daerah kritis:  $Z > 1.96$  atau  $Z < -1.96$ ")

```

```

val_list = df["SumbuKecil"].head(20).values.tolist()
tset, pval = stats.ttest_1samp(val_list, 50)
print("Nilai P: ", pval)
if pval < 0.05:
    print("Pvalue lebih kecil dari alpha, Hipotesis null tidak diterima,
    ↳ sehingga rata-rata bukan 50")
else:
    print("Pvalue lebih besar dari alpha, Hipotesis null diterima, sehingga
    ↳ rata-rata 50")
df.boxplot(column="SumbuKecil")

```

4c. Nilai Rata-rata 20 baris pertama kolom Sumbu Kecil bukan 50?

h_0 : rata-rata = 50

h_1 : rata-rata \neq 50

$\alpha = 0.05$

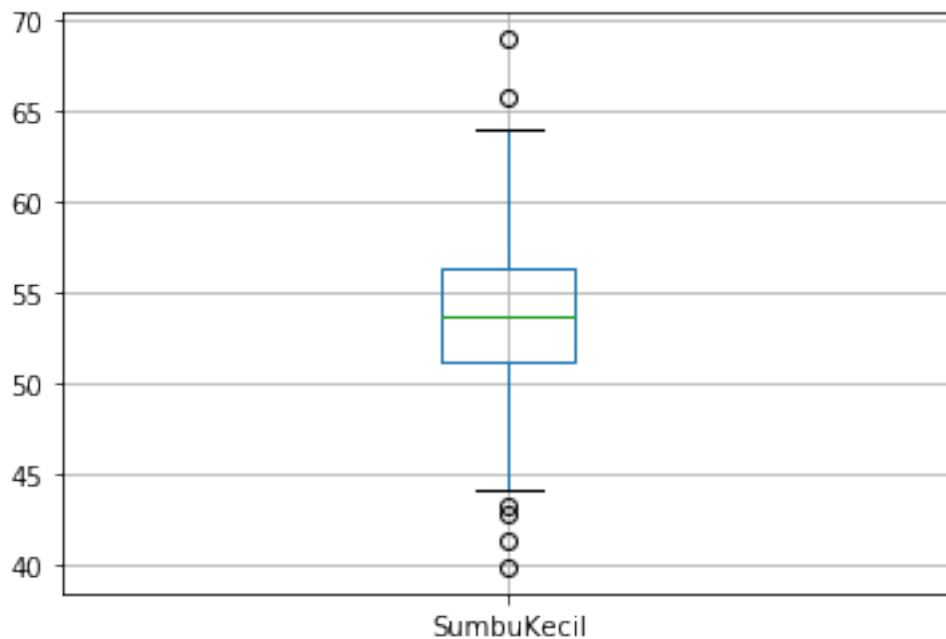
Tes yang digunakan: One Sample Two-Tailed Normal test

Daerah kritis: $Z > 1.96$ atau $Z < -1.96$

Nilai P: 3.3020185644245e-06

Pvalue lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga rata-rata bukan 50

[71]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca50ae48>



```

[72]: print("4d. Proporsi nilai Diameter yang lebih dari 85, adalah tidak sama dengan
    ↳ 15% ?")
print("h0: p = 0.15")

```

```

print("h1: p != 0.15")
print("alpha = 0.05")
print("Tes yang digunakan: Two-Tailed Binomial Test")
print("Daerah kritis: Z > 1.96 atau Z < -1.96")
dia_above_85 = len(df[df['Diameter'] > 85])
pval = stats.binom_test(dia_above_85, len(df), 0.15)
print("Nilai P: ", pval)
if pval < 0.05:
    print("Pvalue lebih kecil dari alpha, Hipotesis null tidak diterima,
    ↳ sehingga proporsi bukan 15%")
else:
    print("Pvalue lebih besar dari alpha, Hipotesis null diterima, sehingga
    ↳ proporsi 15%")
df.boxplot(column="Diameter")

```

4d. Proporsi nilai Diameter yang lebih dari 85, adalah tidak sama dengan 15% ?

$H_0: p = 0.15$

$H_1: p \neq 0.15$

$\alpha = 0.05$

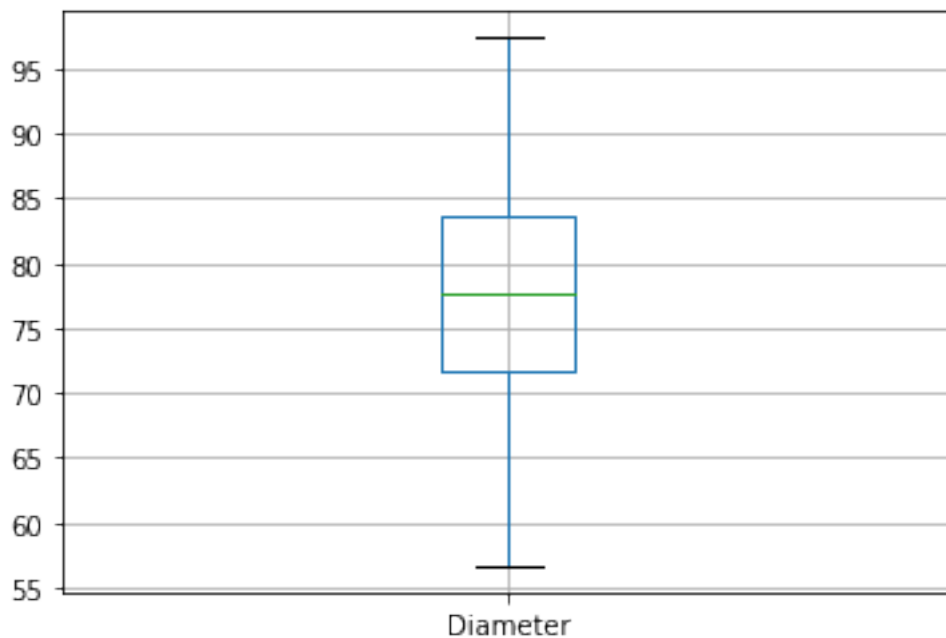
Tes yang digunakan: Two-Tailed Binomial Test

Daerah kritis: $Z > 1.96$ atau $Z < -1.96$

Nilai P: 0.0070570916667948

Pvalue lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga proporsi bukan 15%

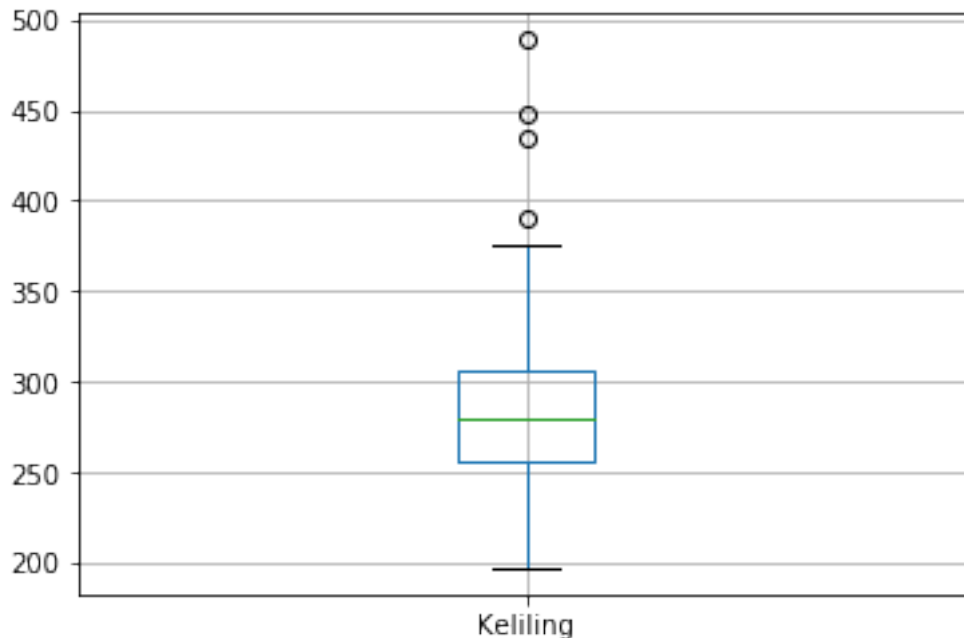
[72]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca5f36c8>



```
[73]: print("4e. Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari 5% ?
      ↪")
print("h0: p = 0.05")
print("h1: p < 0.05")
print("alpha = 0.05")
print("Tes yang digunakan: One-Tailed Binomial Test")
print("Daerah kritis: Z < -1.645")
kel_below_100 = len(df[df['Keliling'] < 100])
print(kel_below_100)
pval = stats.binom_test(kel_below_100, len(df), 0.05)
print("Nilai P: ", pval)
# tes yang diperlukan one-tailed, sementara fungsi menghitung two-ended, ↪
↪sehingga P daerah kritis perlu 2x lipat
# note: keberadaan tail kedua dinilai tidak memengaruhi hasil
if pval < 0.10:
    print("Pvalue lebih kecil dari alpha, Hipotesis null tidak diterima, ↪
    ↪sehingga proporsi bukan 5%")
else:
    print("Pvalue lebih besar dari alpha, Hipotesis null diterima, sehingga ↪
    ↪proporsi 5%")
df.boxplot(column="Keliling")
```

```
4e. Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari 5% ?
h0: p = 0.05
h1: p < 0.05
alpha = 0.05
Tes yang digunakan: One-Tailed Binomial Test
Daerah kritis: Z < -1.645
0
Nilai P: 1.7481550525507304e-11
Pvalue lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga proporsi
bukan 5%
```

```
[73]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca66db88>
```



```
[74]: # 5.a
print("5a. Data kolom AreaBulatan dibagi 2 sama rata: bagian awal dan bagian_
      ↳akhir kolom. Benarkah rata-rata kedua bagian tersebut sama?")
df_length = int(len(df)/2)
sample1 = df[df_length:]["AreaBulatan"]
sample2 = df[:df_length]["AreaBulatan"]
print("1. h0 : meanSample1 = meanSample2")
print("2. h1 : meanSample1 != meanSample2")
print("3. alpha = 0.05")
print("4. Two-Sample Two-Tailed Normal Test")
print("Daerah kritis: Z > 1.96 atau Z < -1.96")
ttest,pval = stats.ttest_ind(sample1, sample2)
print("5. Nilai P: ", pval)
if pval < 0.05:
    print("6. P lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga_
          ↳rata-rata kedua bagian tidak sama")
else:
    print("6. P lebih besar dari alpha, Hipotesis null diterima, sehingga_
          ↳rata-rata kedua bagian tidak sama")
df.boxplot(column="AreaBulatan")
```

5a. Data kolom AreaBulatan dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata kedua bagian tersebut sama?

1. h0 : meanSample1 = meanSample2
2. h1 : meanSample1 != meanSample2
3. alpha = 0.05

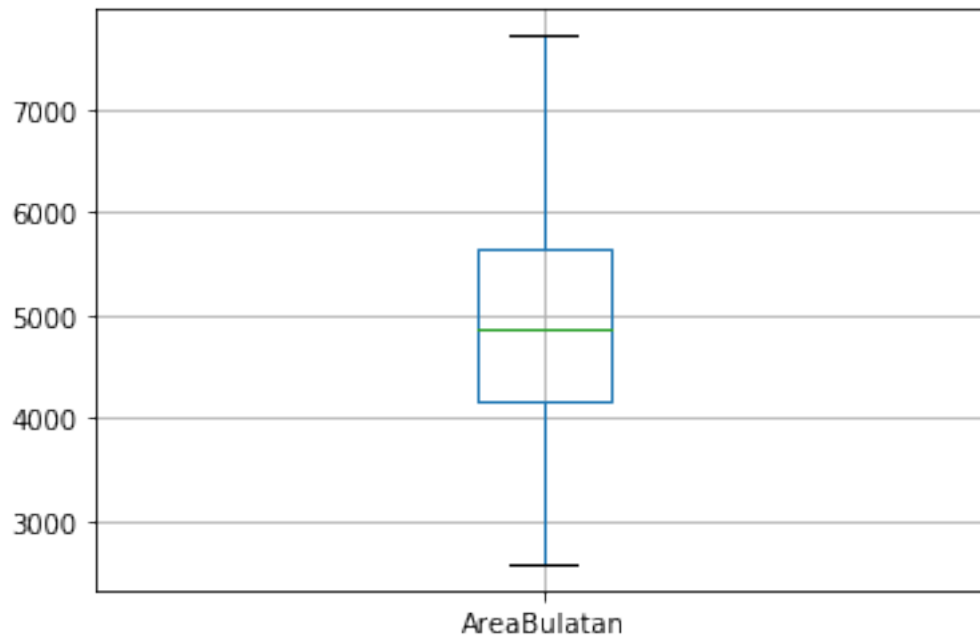
4. Two-Sample Two-Tailed Normal Test

Daerah kritis: $Z > 1.96$ atau $Z < -1.96$

5. Nilai P: 1.6612692497332565e-51

6. P lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga rata-rata kedua bagian tidak sama

[74]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca6db4c8>



```
[75]: #5.b
print("5b. Data kolom Kadar Air dibagi 2 sama rata: bagian awal dan bagian_
      ↳akhir kolom. Benarkah rata-rata bagian awal lebih besar dari pada bagian_
      ↳akhir sebesar 0.2?")
df_length = int(len(df)/2)
sample1 = df[df_length:]["KadarAir"]
sample2 = df[:df_length]["KadarAir"]
print("1. h0 : meanSample1-meanSample2 = 2")
print("2. h1 : meanSample1-meanSample2 > 2")
print("3. alpha = 0.05")
print("4. Two-Sample Two-Tailed Normal Test")
print("Daerah kritis: Z > 1.645")
ttest,pval = stats.ttest_ind(sample1, sample2)
print("5. Nilai P: ", pval)
if pval < 0.05:
    print("6. P lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga_
          ↳perbedaan rata-rata kedua bagian sama dengan dua")
else:
```

```
print("6. P lebih besar dari alpha, Hipotesis null diterima, sehingga
↳perbedaan rata-rata kedua bagian tidak sama dengan dua")
df.boxplot(column="KadarAir")
```

5b. Data kolom Kadar Air dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata bagian awal lebih besar dari pada bagian akhir sebesar 0.2?

1. h_0 : $\text{meanSample1} - \text{meanSample2} = 2$

2. h_1 : $\text{meanSample1} - \text{meanSample2} > 2$

3. $\alpha = 0.05$

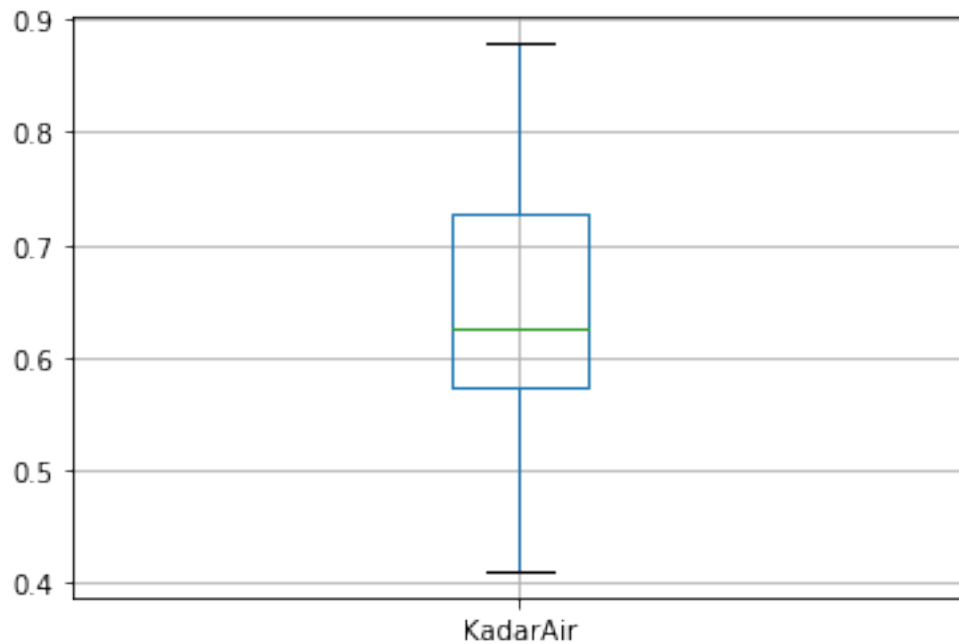
4. Two-Sample Two-Tailed Normal Test

Daerah kritis: $Z > 1.645$

5. Nilai P: 0.0026878582871855895

6. P lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga perbedaan rata-rata kedua bagian sama dengan dua

[75]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca73b948>



```
[76]: # 5.c
print("5c. Rata-rata 20 baris pertama kolom Bulatan sama dengan 20 baris
↳terakhirnya?")
sample1 = df[:20]["Bulatan"]
sample2 = df[-20:]["Bulatan"]
print("1.  $h_0$  :  $\text{meanSample1} = \text{meanSample2}$ ")
print("2.  $h_1$  :  $\text{meanSample1} \neq \text{meanSample2}$ ")
print("3.  $\alpha = 0.05$ ")
print("4. Two-Sample Two-Tailed Normal Test")
```



```

print("Daerah kritis: Z > 1.96 atau Z < -1.96")
ttest,pval = stats.ttest_ind(sample1, sample2)
print("5. Nilai P: ", pval)
if pval < 0.05:
    print("6. P lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga
    ↪rata-rata kedua bagian tidak sama")
else:
    print("6. P lebih besar dari alpha, Hipotesis null diterima, sehingga
    ↪rata-rata kedua bagian sama")
df.boxplot(column="Bulatan")

```

5c. Rata-rata 20 baris pertama kolom Bulatan sama dengan 20 baris terakhirnya?

1. H_0 : $\text{meanSample1} = \text{meanSample2}$

2. H_1 : $\text{meanSample1} \neq \text{meanSample2}$

3. $\alpha = 0.05$

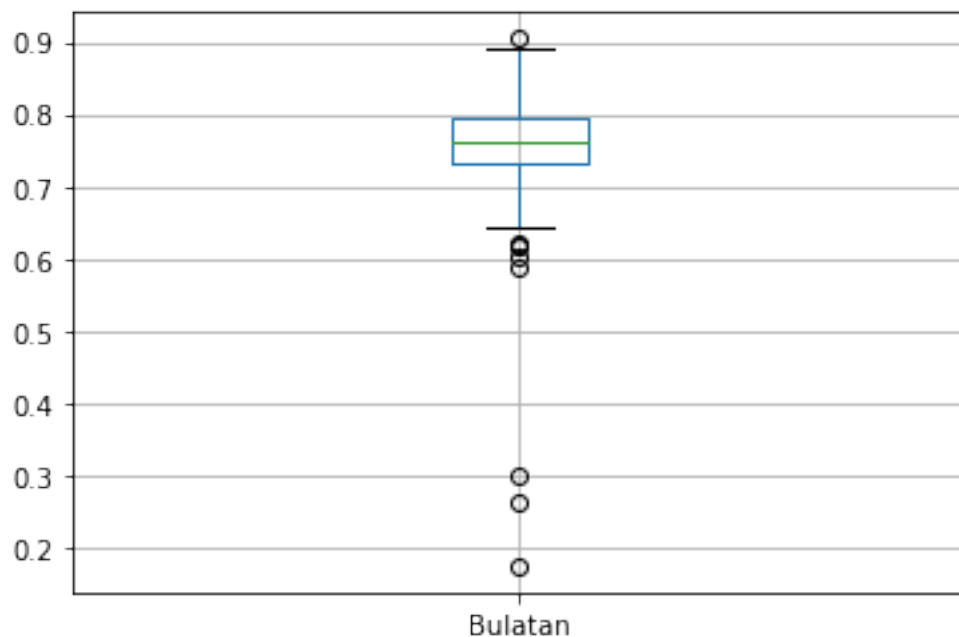
4. Two-Sample Two-Tailed Normal Test

Daerah kritis: $Z > 1.96$ atau $Z < -1.96$

5. Nilai P: 0.0016919037522060854

6. P lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga rata-rata kedua bagian tidak sama

[76]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca791588>



```

[77]: #5.d
print("5d. Proporsi nilai bagian awal Ransum yang lebih dari 2, adalah lebih
    ↪besar daripada, proporsi nilai yang sama di bagian akhir Ransum?")

```

```

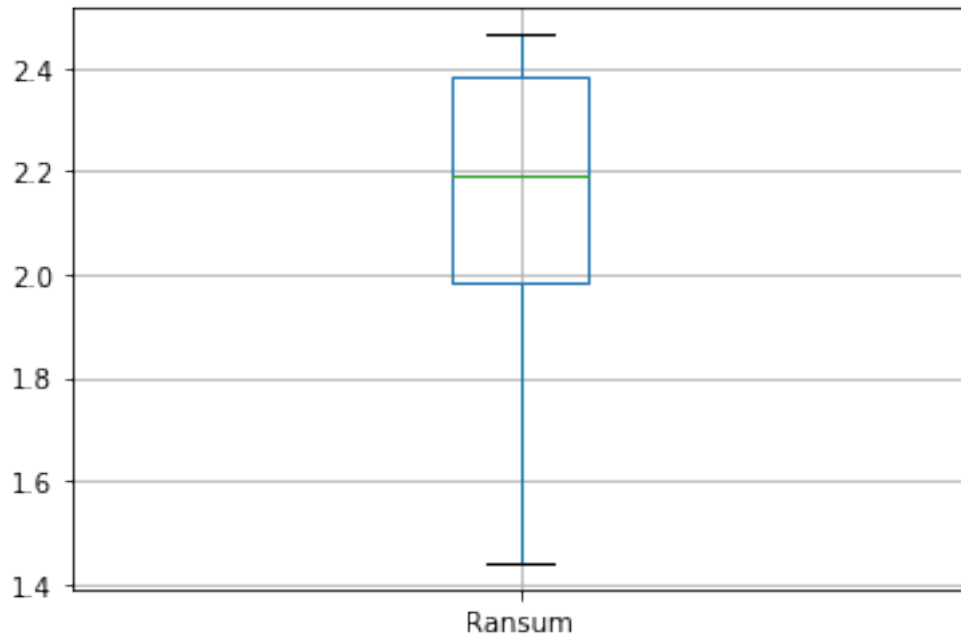
data = df[df.Ransum > 2]["Ransum"]
data_length = int(len(data)/2)
sample1 = data[df_length:]
sample2 = data[:df_length]
print("1. h0 : Sample1 = Sample2")
print("2. h1 : Sample1 > Sample2")
print("3. alpha = 0.05")
pt = len(sample1)/len(df)
pc = len(sample2)/len(df)
print("4. Two-Sample One-Tailed Binomial Test")
print("Daerah kritis: Z > 1.645")
p = (len(sample1) + len(sample2))/len(df)
q = 1-p
z = (pt-pc)/math.sqrt(p * q * ((1/len(sample1) + (1/len(sample2)))))
pval = stats.norm.sf(abs(z)) #one-sided
print("5. Nilai P: ", pval)
if pval < 0.05:
    print("6. P lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga
    ↳proporsi bagian awal lebih besar")
else:
    print("6. P lebih besar dari alpha, Hipotesis null diterima, sehingga
    ↳proporsi bagian awal tidak lebih besar")
df.boxplot(column="Ransum")

```

5d. Proporsi nilai bagian awal Ransum yang lebih dari 2, adalah lebih besar daripada, proporsi nilai yang sama di bagian akhir Ransum?

1. h0 : Sample1 = Sample2
2. h1 : Sample1 > Sample2
3. alpha = 0.05
4. Two-Sample One-Tailed Binomial Test
- Daerah kritis: Z > 1.645
5. Nilai P: 3.378725067634258e-08
6. P lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga proporsi bagian awal lebih besar

[77]: <matplotlib.axes._subplots.AxesSubplot at 0x15aca78d4c8>



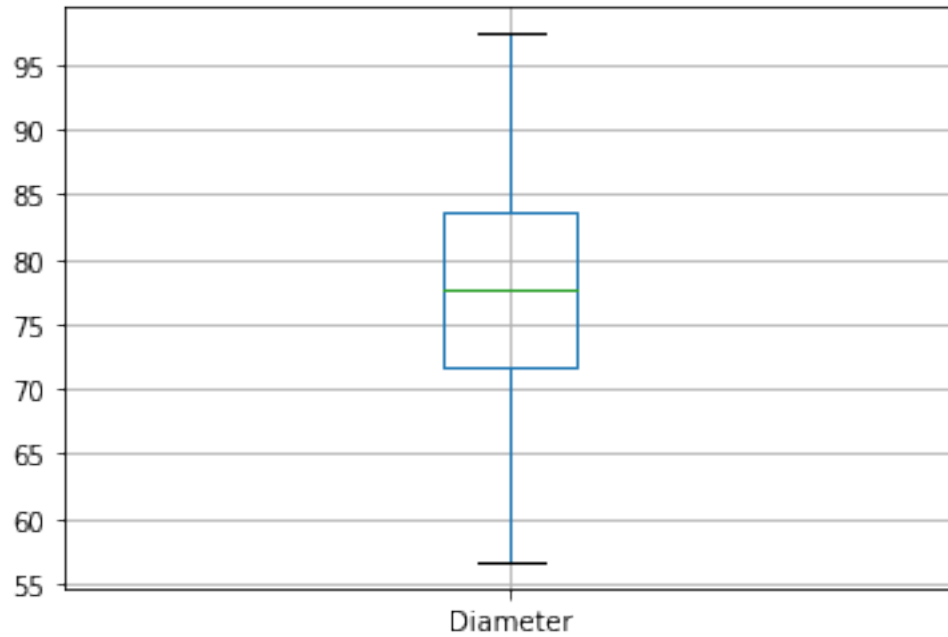
```
[79]: #5.e
print("5e. Bagian awal kolom Diameter memiliki variansi yang sama dengan bagian_
      ↳akhirnya?")
data = df[df.Ransum > 2]["Diameter"]
data_length = int(len(data)/2)
sample1 = data[df_length:]
sample2 = data[:df_length]
print("1. h0 : VarSample1 = VarSample2")
print("2. h1 : VarSample1 != VarSample2")
print("3. alpha = 0.05")
print("4. Two-Tailed Dist f Test")
F = sample1.var() / sample2.var()
p_val = stats.f.cdf(F, len(sample1)-1, len(sample2)-1)
print("5. Nilai P: ", pval)
if pval < 0.05:
    print("6. P lebih kecil dari alpha, Hipotesis null tidak diterima, sehingga_
          ↳variansi tidak sama")
else:
    print("6. P lebih besar dari alpha, Hipotesis null diterima, sehingga_
          ↳variansi sama")
df.boxplot(column="Diameter")
```

5e. Bagian awal kolom Diameter memiliki variansi yang sama dengan bagian akhirnya?

1. $h_0 : \text{VarSample1} = \text{VarSample2}$
2. $h_1 : \text{VarSample1} \neq \text{VarSample2}$

3. $\alpha = 0.05$
4. Two-Tailed Dist f Test
5. Nilai P: $3.378725067634258e-08$
6. P lebih kecil dari α , Hipotesis null tidak diterima, sehingga variansi tidak sama

[79]: <matplotlib.axes._subplots.AxesSubplot at 0x15ac828fa48>



[]: