

K01-T1-IF2220-13519026

April 15, 2021

## 1 Tugas Besar IF2220 Probabilitas dan Statistika

## 2 Penarikan Kesimpulan dan Pengujian Hipotesis

### 2.0.1 Tujuan:

- Mahasiswa memahami dan dapat menyelesaikan persoalan distribusi peluang variabel random diskrit dan kontinu, dan
- Mahasiswa mampu menyelesaikan persoalan untuk menarik kesimpulan mengenai parameter populasi yang diperoleh dari data hasil eksperimen.
- Mahasiswa mampu menyelesaikan persoalan pengujian hipotesis.

### 2.0.2 Kelas : K01

### 2.0.3 Dibuat oleh :

1. Gde Anantha Priharsena - 13519026
2. Reihan Andhika Putra - 13519043

## 3 Section 0 - Persiapan

### 3.1 Import Library

Import library yang akan digunakan

```
[1]: import pandas as pd
import numpy as np
from scipy import stats as st
import matplotlib.pyplot as plt
import seaborn as sns
import math
%matplotlib inline
```

### 3.2 Pembacaan Data

Pembacaan data dari file eksternal dan pemberian untuk setiap nama kolom.

```
[2]:
```

```
column = [
    "id", "Daerah", "SumbuUtama", "SumbuKecil", "Keunikan", "AreaBulatan", "Diameter",
    "KadarAir", "Keliling", "Bulatan", "Ransum", "Kelas"]
df = pd.read_csv('Gandum.csv', names = column)
df
```

```
[2]:
```

	id	Daerah	SumbuUtama	SumbuKecil	Keunikan	AreaBulatan	Diameter	\
0	1	5781	128.288875	58.470846	0.890095	5954	85.793926	
1	2	4176	109.348294	49.837688	0.890098	4277	72.918093	
2	3	4555	114.427991	52.151207	0.890105	4706	76.155145	
3	4	4141	108.701190	49.457349	0.890499	4236	72.611879	
4	5	5273	122.747868	55.757848	0.890876	5431	81.937733	
..	...	...	...	...	...	...	...	
495	496	5083	120.083450	54.821580	0.889709	5179	80.447975	
496	497	4432	112.367050	51.294914	0.889726	4550	75.119889	
497	498	5020	119.873742	54.718545	0.889740	5104	79.947874	
498	499	4035	107.311728	48.930802	0.889996	4150	71.676506	
499	500	3379	99.014789	44.631551	0.892647	3491	65.591741	

	KadarAir	Keliling	Bulatan	Ransum	Kelas
0	0.674090	316.756	0.724041	2.194066	1
1	0.596231	260.346	0.774227	2.194088	1
2	0.776641	279.606	0.732159	2.194158	1
3	0.633180	260.478	0.766961	2.197877	1
4	0.669842	302.730	0.723031	2.201446	1
..	...	...	...	...	...
495	0.534827	286.377	0.778850	2.190441	2
496	0.601194	270.823	0.759344	2.190608	2
497	0.528421	285.799	0.772311	2.190733	2
498	0.584698	258.503	0.758791	2.193132	2
499	0.653578	237.593	0.752196	2.218493	2

[500 rows x 12 columns]

## 4 Section 1 - Descriptive Statistics

Menulis deskripsi statistika (Descriptive Statistics) dari semua kolom pada data yang bersifat numerik, terdiri dari mean, median, modus, standar deviasi, variansi, range, nilai minimum, maksimum, kuartil, IQR, skewness dan kurtosis. Perhitungan nilai kurtosis menggunakan definisi Pearson. Ditambahkan juga perhitungan jumlah data untuk setiap kolom.

```
[3]: # Prosedur untuk menulis deskripsi statistik dari suatu kolom di tabel
def printStatistic(Column):
    # Tulis statistik deskriptif dari kolom Column
    print("count\t\t\t: {}".format(len(Column)))
    print("mean\t\t\t: {}".format(Column.mean()))
    print("median\t\t\t: {}".format(Column.median()))
```

```

print("modus\t\t\t: {}".format(', '.join(map(str, Column.mode()))))
print("standar deviasi\t\t: {}".format(Column.std()))
print("variansi\t\t: {}".format(Column.std()**2))
print("range\t\t\t: {}".format(Column.max()-Column.min()))
print("nilai minimum\t\t: {}".format(Column.min()))
print("nilai maksimum\t\t: {}".format(Column.max()))
quartiles = Column.quantile([0,0.25,0.5,0.75,1])
for i,quartile in enumerate(quartiles):
    print("quartil-{}\t\t: {}".format(i,quartile))
print("IQR\t\t\t: {}".format(quartiles[0.75]-quartiles[0.25]))
print("skewness\t\t: {}".format(Column.skew()))
print("kurtosis\t\t: {}".format(st.kurtosis(Column, fisher=False)))

```

#### 4.1 1. id

Id tidak perlu ditulis deskripsi statistiknya karena id merepresentasikan identifier yang unik untuk tiap ransum.

#### 4.2 2. Daerah

```

[4]: # Ambil kolom daerah
Daerah = df["Daerah"]
# Tulis statistik deskriptif dari kolom daerah
print("Statistik deskriptif kolom daerah")
printStatistic(Daerah)

```

```

Statistik deskriptif kolom daerah
count                : 500
mean                 : 4801.246
median               : 4735.0
modus                : 3992, 4881, 5642, 6083
standar deviasi      : 986.3954914816017
variansi             : 972976.0656152305
range                : 4931
nilai minimum        : 2522
nilai maksimum       : 7453
quartil-0            : 2522.0
quartil-1            : 4042.75
quartil-2            : 4735.0
quartil-3            : 5495.5
quartil-4            : 7453.0
IQR                  : 1452.75
skewness              : 0.23814408738280812
kurtosis              : 2.557727571090344

```

### 4.3 3. SumbuUtama

```
[5]: # Ambil kolom SumbuUtama
SumbuUtama = df["SumbuUtama"]
# Tulis statistik deskriptif dari kolom SumbuUtama
print("Statistik deskriptif kolom SumbuUtama")
printStatistic(SumbuUtama)
```

Statistik deskriptif kolom SumbuUtama

```
count          : 500
mean           : 116.04517136778
median         : 115.40513995
modus          : 74.13311404, 74.36402121, 74.69188071, 76.2931638,
76.78904267, 77.03362772, 77.04768244, 77.09079003, 77.41707342, 81.27755239,
81.3042004, 82.57014058, 82.87781414, 83.37848692, 83.82327008, 84.05848002,
84.32356418, 84.54078476, 84.58114773, 85.12478457, 86.56781638, 86.95241143,
87.3845036, 87.41286344, 87.97830505, 87.98665592, 88.0017938, 88.02937756,
88.11593906, 88.77990424, 88.92210563, 89.40212268, 89.49579817, 89.54280748,
90.07527599, 90.76577027, 91.05330784, 91.13033539, 91.40038634, 91.87618278,
92.20828955, 92.21294021, 92.2293165, 92.43690606, 92.47994738, 92.58932307,
93.52974726, 93.5356072, 93.76361246, 93.9988852, 94.00143053, 94.00787826,
94.12263105, 94.17620603, 94.26939767, 94.53973823, 94.59943367, 94.63650283,
94.91539197, 95.00364795, 95.17012555, 95.2470937, 95.37622831, 95.46121147,
95.69548849, 95.74362731, 95.97894614, 96.03513343, 96.14211158, 96.23547918,
96.29743756, 96.33549432, 96.35268965, 96.40313032, 96.60035011, 97.08081574,
97.32531129, 97.47084902, 97.69259544, 97.77562771, 97.97380616, 98.21898138,
98.2198269, 98.35854914, 98.37247663, 98.37593738, 98.86216016, 99.01478925,
99.21752118, 99.32051276, 99.47215247, 99.49226615, 99.60332963, 99.79394795,
99.88402006, 99.93450497, 99.93602686, 100.0223281, 100.4167531, 100.5093206,
100.5480488, 100.698893, 100.7357227, 100.7501023, 101.1105249, 101.3157423,
101.45421, 101.5810398, 101.8422762, 101.9698018, 101.986251, 102.1971523,
102.2953523, 102.320702, 102.3709892, 102.4485604, 102.4749007, 102.8634748,
103.2901421, 103.5232905, 103.6999129, 103.7527928, 103.8253162, 103.894842,
103.9579819, 104.1688036, 104.1744438, 104.2664186, 104.272109, 104.3276353,
104.3280876, 104.3378697, 104.3388145, 104.6666913, 104.8104398, 104.9579846,
105.3373586, 105.4041588, 105.5405362, 105.5612967, 105.6187899, 105.6400951,
105.6451601, 105.9127833, 106.0365044, 106.1024587, 106.1158346, 106.2162594,
106.25667, 106.2923105, 106.3301589, 106.3932719, 106.4473886, 106.5010619,
106.5811347, 106.5972033, 106.7211415, 106.8064769, 106.8208342, 106.8561506,
106.9162059, 106.9605737, 106.9693936, 107.0147691, 107.1803967, 107.2441145,
107.2664178, 107.2992964, 107.3117282, 107.3323015, 107.4234632, 107.4901282,
107.515859, 107.6060473, 107.6272465, 107.780575, 107.8182786, 107.830578,
107.8974401, 107.9361607, 108.0411686, 108.0691562, 108.0842945, 108.1493814,
108.3345087, 108.6018895, 108.6173366, 108.7011905, 108.9832503, 109.0780728,
109.0975949, 109.2221888, 109.2448278, 109.3320256, 109.3482943, 109.3557051,
109.3909238, 109.419762, 109.7155482, 109.7356066, 109.8055701, 109.8486599,
110.1418199, 110.2029953, 110.4464398, 110.4531344, 110.4948924, 110.5368768,
110.6219338, 110.6894983, 110.7022218, 110.9722142, 111.0079213, 111.231913,
```

111.2556534, 111.3236714, 111.7058775, 111.7161753, 111.7569232, 111.7977586,  
112.1203439, 112.173828, 112.3670498, 112.4289564, 113.0192872, 113.0268924,  
113.0982598, 113.3293145, 113.3307374, 113.4990733, 113.5134456, 113.529303,  
113.5868234, 113.664381, 113.719943, 113.7689435, 113.8160966, 113.8714491,  
114.015075, 114.1212737, 114.2651167, 114.4279914, 114.4543656, 114.6402584,  
114.7953243, 114.9422326, 114.9435821, 115.0649959, 115.2274784, 115.3967443,  
115.4135356, 115.5183228, 115.7148758, 115.7543147, 115.8597799, 115.8966117,  
116.0601546, 116.1939725, 116.2937463, 116.3895145, 116.4064279, 116.4812077,  
117.2504881, 117.3431374, 117.4265099, 117.4406074, 117.7277342, 117.9279205,  
118.0428891, 118.0977083, 118.2361891, 118.2847646, 118.5822997, 118.805766,  
118.8225877, 118.8853189, 118.9054023, 119.0035819, 119.0063356, 119.2490531,  
119.4916115, 119.6187778, 119.8611577, 119.8737416, 119.9345923, 120.0379792,  
120.0437924, 120.0834497, 120.1209915, 120.1405384, 120.6979911, 120.7283328,  
120.7373788, 120.8171894, 120.9747517, 121.0072161, 121.0657867, 121.3226446,  
121.5526062, 121.9966621, 122.0421307, 122.0585037, 122.1619579, 122.1912997,  
122.2136145, 122.7478685, 122.8211457, 123.0018106, 123.0033869, 123.0370935,  
123.0789249, 123.1084563, 123.1519679, 123.4300431, 123.4962215, 123.4969179,  
123.5226615, 123.6655012, 123.6906866, 123.8644522, 123.9306004, 123.9435304,  
123.9603486, 123.991228, 124.0033857, 124.084748, 124.2348815, 124.2806183,  
124.4496928, 124.5148772, 124.6260959, 124.7248074, 124.8235842, 124.904374,  
124.9396755, 125.003672, 125.1644596, 125.2441841, 125.2643871, 125.3132385,  
125.3959043, 125.3970223, 125.5554308, 125.5654589, 125.5864509, 125.6229894,  
125.8384269, 125.8670623, 125.9189141, 125.9803048, 126.2778934, 126.2920083,  
126.3149145, 126.4068098, 126.4776571, 126.6010653, 126.6443526, 126.8097594,  
127.0139951, 127.0159877, 127.1641483, 127.2987968, 127.5101183, 127.6522495,  
128.0022172, 128.0663147, 128.1784256, 128.2147081, 128.2888753, 128.4823393,  
128.6306201, 128.6889548, 128.7618704, 129.0205405, 129.037623, 129.0742991,  
129.0803, 129.0807906, 129.098407, 129.1812458, 129.1862369, 129.5767591,  
129.7610596, 129.8095836, 130.0598902, 130.2207659, 130.2621314, 130.2739254,  
130.3923317, 130.4379451, 130.4457155, 130.6999931, 130.83002, 130.9221608,  
130.9643735, 131.152746, 131.2915779, 131.4621525, 131.609712, 131.629047,  
131.6302887, 131.8083893, 132.2525025, 132.3942597, 132.4816506, 132.6956228,  
132.723628, 132.7560217, 132.7895598, 133.0157281, 133.084582, 133.1994688,  
133.2444458, 133.3679075, 133.4361583, 133.4399856, 133.4634273, 133.5949267,  
133.6271788, 133.6566273, 133.881794, 133.9713999, 134.1673184, 134.2316812,  
134.2851441, 134.4606566, 134.4930235, 134.5195056, 134.5617589, 134.6662091,  
134.8275164, 134.9710615, 135.1197917, 135.2490597, 135.2677147, 135.8156503,  
135.8597438, 135.9417416, 136.0519648, 136.3036028, 136.3800864, 136.5709394,  
136.6145396, 136.6195076, 136.7764526, 136.8780111, 136.8870215, 137.0297517,  
137.1638147, 137.4244119, 137.8932058, 137.9956035, 138.0376883, 138.2095558,  
138.3324882, 138.5717146, 139.0147323, 139.0922395, 139.180131, 139.2741572,  
139.5753817, 139.9708612, 140.3828578, 140.4063965, 140.4275867, 141.0341403,  
141.0651834, 141.1236288, 141.1434891, 141.3583591, 141.4124755, 141.7730175,  
142.0806537, 142.34725, 142.3807955, 142.4027871, 142.5335656, 142.9495008,  
143.9124976, 144.267473, 144.3587743, 144.4331141, 144.4487392, 144.9073336,  
146.177056, 146.3717849, 146.5902286, 146.6962997, 146.7290182, 147.0057745,  
147.138743, 147.4293134, 147.8905453, 150.429964, 151.6017657, 152.0684401,  
152.1134913, 153.583387, 227.1054624, 227.9285827

```

standar deviasi      : 18.28262595755935
variansi             : 334.25441190402296
range                : 153.79546865999998
nilai minimum        : 74.13311404
nilai maksimum       : 227.9285827
kuartil-0            : 74.13311404
kuartil-1            : 104.116098175
kuartil-2            : 115.40513995
kuartil-3            : 129.046792025
kuartil-4            : 227.9285827
IQR                  : 24.930693849999997
skewness             : 0.7615287378076652
kurtosis             : 7.275373247250751

```

#### 4.4 4. SumbuKecil

```

[6]: # Ambil kolom SumbuKecil
SumbuKecil = df["SumbuKecil"]
# Tulis statistik deskriptif dari kolom SumbuKecil
print("Statistik deskriptif kolom SumbuKecil")
printStatistic(SumbuKecil)

```

Statistik deskriptif kolom SumbuKecil

```

count      : 500
mean       : 53.71524598896
median     : 53.731198774999996
modus      : 39.90651744, 41.43641868, 42.87187941, 43.28497882,
44.11935531, 44.12313831, 44.55523434, 44.63155114, 44.80273426, 44.81011044,
44.87519946, 44.94078546, 45.24113928, 45.26165673, 45.3742466, 45.52911212,
45.78398888, 45.88286667, 45.89775233, 46.01282025, 46.05036728, 46.23892946,
46.81157756, 46.83791704, 46.94326548, 46.99201881, 47.02590414, 47.12533466,
47.13315533, 47.13968338, 47.15593922, 47.31982512, 47.54017615, 47.64940038,
47.69255121, 47.73887538, 47.9325942, 48.03406538, 48.05318758, 48.07414422,
48.12622906, 48.16081418, 48.27920161, 48.33770705, 48.44034455, 48.55849663,
48.62133126, 48.73552413, 48.8009917, 48.80975228, 48.83939057, 48.84442145,
48.90235887, 48.93080224, 48.95934176, 49.06249966, 49.06935374, 49.10437226,
49.12511552, 49.14862456, 49.16736189, 49.22548543, 49.24267814, 49.2484593,
49.31615584, 49.33311864, 49.3411712, 49.34837868, 49.37002437, 49.39653936,
49.44130516, 49.45734894, 49.52659704, 49.58314468, 49.64746533, 49.66997344,
49.75342248, 49.8376875, 49.88557273, 49.89166435, 49.99383017, 50.02465547,
50.13791447, 50.15451213, 50.18648304, 50.20161732, 50.21689738, 50.23674047,
50.26503783, 50.30696932, 50.31824228, 50.34002156, 50.3440607, 50.35078641,
50.38129497, 50.40674957, 50.43880144, 50.44282234, 50.49647509, 50.51191604,
50.51491056, 50.52519208, 50.52718252, 50.55859111, 50.57754473, 50.62414147,
50.6631383, 50.70506339, 50.71577122, 50.74565115, 50.79059698, 50.79355129,
50.8078401, 50.81390793, 50.87502689, 50.90915187, 50.9543441, 51.01698501,
51.05518917, 51.07067636, 51.07711152, 51.08260448, 51.14681966, 51.18395809,
51.19004627, 51.19475302, 51.20220775, 51.28416875, 51.29491408, 51.31237346,

```

51.35134765, 51.36408785, 51.36464535, 51.36513446, 51.36922513, 51.3813348,  
51.40045446, 51.41678314, 51.46334912, 51.48838799, 51.4952914, 51.5104279,  
51.52527338, 51.53272243, 51.56715505, 51.59419579, 51.64579845, 51.66611058,  
51.66774487, 51.68537529, 51.71361378, 51.71568655, 51.78753009, 51.85164606,  
51.86429927, 51.87272624, 51.87325399, 51.90747401, 51.91358009, 51.92222212,  
51.92848667, 51.93962023, 51.9735486, 51.99738757, 52.0250802, 52.02727997,  
52.04349114, 52.06616637, 52.0746236, 52.07642841, 52.09462225, 52.10565058,  
52.14737228, 52.15120679, 52.17637903, 52.19604698, 52.22960782, 52.25358497,  
52.27703901, 52.30281469, 52.30336252, 52.31378371, 52.31804963, 52.32413559,  
52.32501448, 52.38486049, 52.40467224, 52.4323366, 52.43244079, 52.45069063,  
52.4511359, 52.47633057, 52.48567756, 52.48729068, 52.49343954, 52.51306281,  
52.5886165, 52.60182365, 52.61238315, 52.61389592, 52.65198942, 52.68167884,  
52.69032762, 52.72642645, 52.73574362, 52.76486452, 52.77875015, 52.78024542,  
52.78100382, 52.81908582, 52.84771577, 52.84775827, 52.88819805, 52.89732251,  
52.90751618, 52.95148583, 52.9861638, 52.99859978, 53.01542506, 53.09012652,  
53.0920322, 53.13975738, 53.15412282, 53.17603697, 53.1912048, 53.1936825,  
53.23867944, 53.28323892, 53.30208661, 53.3142954, 53.32667827, 53.34879907,  
53.36725831, 53.38323965, 53.38772664, 53.40982683, 53.42470023, 53.43802216,  
53.44922494, 53.45390584, 53.45686204, 53.47694772, 53.48524695, 53.53444948,  
53.54299328, 53.56818644, 53.63079347, 53.70860038, 53.7163328, 53.72745253,  
53.73494502, 53.74148331, 53.76839227, 53.80876967, 53.81889593, 53.84724626,  
53.85539635, 53.86276766, 53.87284326, 53.87480874, 53.8901512, 53.91266267,  
53.97675659, 53.99067461, 54.05545555, 54.05568063, 54.08801741, 54.10522429,  
54.21520552, 54.2487085, 54.28666365, 54.30216853, 54.3029021, 54.36164284,  
54.37499282, 54.37937746, 54.38703038, 54.4026735, 54.41165575, 54.4259274,  
54.4293035, 54.43345386, 54.45602198, 54.48061344, 54.48500011, 54.49081793,  
54.50369237, 54.52957006, 54.53488492, 54.55141565, 54.57328405, 54.66357193,  
54.66500185, 54.67086487, 54.71854525, 54.73130459, 54.73755333, 54.7883112,  
54.81120375, 54.82157996, 54.82410069, 54.83162027, 54.83454494, 54.86535794,  
54.92936825, 54.92957964, 54.94802805, 54.97468691, 55.03698356, 55.03706832,  
55.04964734, 55.08209826, 55.08589466, 55.11140291, 55.19301256, 55.20183645,  
55.21937217, 55.22951728, 55.27701192, 55.28355112, 55.30187602, 55.31763147,  
55.33510202, 55.34456288, 55.34858972, 55.35515604, 55.37329946, 55.38850592,  
55.39145755, 55.39471737, 55.39486495, 55.41306133, 55.41384666, 55.44460968,  
55.44610788, 55.46048507, 55.47569683, 55.50969084, 55.57226815, 55.61545032,  
55.62897901, 55.63227728, 55.66036397, 55.6734017, 55.68618329, 55.72407565,  
55.73154196, 55.73978445, 55.75784788, 55.7786501, 55.81294139, 55.81780631,  
55.81961637, 55.86103791, 55.8747349, 55.87488082, 55.91659689, 55.91833049,  
55.92340017, 55.95378028, 55.96028775, 55.96674868, 55.96797915, 55.97868866,  
56.00515296, 56.11190573, 56.11813541, 56.14063497, 56.14671184, 56.19223391,  
56.20142521, 56.20161967, 56.23539815, 56.29384365, 56.32429689, 56.32774126,  
56.36935775, 56.37402054, 56.38621395, 56.39083506, 56.3920177, 56.4069029,  
56.41101106, 56.42123689, 56.43017494, 56.44476861, 56.46244162, 56.4790283,  
56.49175078, 56.55003031, 56.60794623, 56.6442029, 56.64723153, 56.6555965,  
56.65960303, 56.66273543, 56.68285423, 56.80713383, 56.84208808, 56.88186745,  
56.88935479, 56.90043992, 56.91197897, 56.94218191, 56.98484397, 56.98886817,  
57.01859044, 57.05443377, 57.06421701, 57.06786443, 57.07943418, 57.17892838,  
57.17941587, 57.23821435, 57.26437529, 57.29556014, 57.37269718, 57.38530573,

```

57.40293611, 57.40544291, 57.43165941, 57.58988352, 57.6131887, 57.61755431,
57.61952234, 57.66596082, 57.70866139, 57.71611459, 57.72255233, 57.73350774,
57.74622982, 57.78159612, 57.90922261, 57.99418631, 58.02999061, 58.04315267,
58.145802, 58.16517707, 58.1835187, 58.21163627, 58.21205006, 58.28623371,
58.3242352, 58.36184935, 58.37701343, 58.45366738, 58.47084588, 58.50998343,
58.54608192, 58.55567096, 58.57195048, 58.78110107, 58.81014204, 58.90687675,
59.0196392, 59.12347743, 59.23182036, 59.254473, 59.31319017, 59.31434447,
59.41414409, 59.47327096, 59.49805441, 59.62625047, 59.71033765, 59.81104559,
59.82281961, 59.84329211, 59.85390462, 59.90157146, 60.15695607, 60.17811905,
60.26176523, 60.41631466, 60.63224136, 60.71394419, 60.7533301, 60.77690138,
60.84233271, 60.86592873, 61.18062105, 61.25272911, 61.31214783, 61.56537661,
61.6740398, 61.75362721, 61.93266284, 62.17534592, 62.21498302, 62.26812033,
62.30497029, 62.34924866, 62.46606562, 62.94989207, 62.96183004, 63.32285358,
63.76230696, 64.01276874, 65.73847541, 68.97769987
standar deviasi      : 4.0710747524750355
variansi             : 16.573649640239672
range                : 29.071182429999993
nilai minimum        : 39.90651744
nilai maksimum       : 68.97769987
kuartil-0            : 39.90651744
kuartil-1            : 51.1935763325
kuartil-2            : 53.731198774999996
kuartil-3            : 56.3251579825
kuartil-4            : 68.97769987
IQR                  : 5.131581650000001
skewness              : -0.010828051555611359
kurtosis              : 3.458850014976625

```

## 4.5 5. Keunikan

```

[7]: # Ambil kolom Keunikan
Keunikan = df["Keunikan"]
# Tulis statistik deskriptif dari kolom Keunikan
print("Statistik deskriptif kolom Keunikan")
printStatistic(Keunikan)

```

Statistik deskriptif kolom Keunikan

```

count      : 500
mean       : 0.8787639143399999
median     : 0.8900454185
modus      : 0.719916226, 0.725552747, 0.731210927, 0.738638695,
0.749281579, 0.752860889, 0.753761573, 0.760664242, 0.761228217, 0.763053057,
0.763177705, 0.764079502, 0.769107427, 0.77428629, 0.776371784, 0.776646824,
0.777504169, 0.786435754, 0.796277419, 0.796930134, 0.800389846, 0.802399543,
0.803559698, 0.804431698, 0.805267574, 0.805682077, 0.807995866, 0.808494981,
0.808851743, 0.810433374, 0.812142038, 0.814795388, 0.815307665, 0.817427302,
0.820087291, 0.820879385, 0.820941139, 0.822925332, 0.822949871, 0.823711598,
0.823715315, 0.826931032, 0.826973165, 0.827466978, 0.827577394, 0.827763047,

```



0.827868744, 0.829087297, 0.831100711, 0.831470984, 0.832023494, 0.832074376,  
0.832303678, 0.832621093, 0.833148913, 0.833989075, 0.834685363, 0.835050434,  
0.836760131, 0.837510773, 0.838102538, 0.838219144, 0.839020612, 0.839088221,  
0.839478002, 0.839530377, 0.839679822, 0.84070239, 0.841344086, 0.841606851,  
0.841881949, 0.842001029, 0.842021957, 0.84206865, 0.842293366, 0.843008304,  
0.843881227, 0.846045387, 0.846387239, 0.846450727, 0.846912546, 0.84734752,  
0.847566309, 0.847935673, 0.848183557, 0.848331996, 0.848355093, 0.849725489,  
0.849977534, 0.850535305, 0.851320389, 0.851417512, 0.851451051, 0.851614663,  
0.852121255, 0.852194755, 0.852463494, 0.85315581, 0.853340239, 0.853508093,  
0.853814335, 0.854631832, 0.855089993, 0.855432012, 0.855470952, 0.856221212,  
0.85668828, 0.856735754, 0.857934894, 0.858005337, 0.858547702, 0.858961293,  
0.859783443, 0.859805283, 0.860485135, 0.861307617, 0.86199911, 0.862052047,  
0.862380367, 0.862592766, 0.86263072, 0.862894078, 0.862924147, 0.862933391,  
0.863666257, 0.863678918, 0.863723389, 0.864028969, 0.86405457, 0.864074307,  
0.865355119, 0.865397365, 0.865503093, 0.865693836, 0.865748729, 0.866080346,  
0.86619073, 0.866243946, 0.866494578, 0.866750215, 0.866962983, 0.867157094,  
0.867244265, 0.867607294, 0.867624752, 0.867839657, 0.867850021, 0.867999767,  
0.868121957, 0.868134, 0.868438765, 0.868702764, 0.869042704, 0.869334971,  
0.869967831, 0.870125573, 0.870212519, 0.870305271, 0.870549219, 0.87074793,  
0.871551294, 0.871853802, 0.872306528, 0.872661936, 0.872735442, 0.872951636,  
0.873000757, 0.873296434, 0.873773432, 0.873910711, 0.874188772, 0.874600106,  
0.874670474, 0.874681096, 0.874726022, 0.874833328, 0.875104544, 0.875258481,  
0.875294022, 0.875405122, 0.875442302, 0.875478285, 0.875690057, 0.875715319,  
0.875725418, 0.875760566, 0.876009151, 0.876207201, 0.876219713, 0.876651614,  
0.876652071, 0.876677165, 0.87689078, 0.877141276, 0.877670279, 0.877788945,  
0.878216546, 0.878572174, 0.87898496, 0.879242202, 0.879323385, 0.879375842,  
0.879423933, 0.879617592, 0.879943617, 0.879944748, 0.879947057, 0.880039868,  
0.880056444, 0.880210999, 0.880286885, 0.880484211, 0.880622003, 0.881030704,  
0.881054947, 0.881591069, 0.881943587, 0.882012553, 0.882050377, 0.882186875,  
0.882333753, 0.882545011, 0.882769548, 0.883116313, 0.883215923, 0.883364255,  
0.883367603, 0.883735602, 0.884513032, 0.885168084, 0.885541309, 0.885642633,  
0.886019582, 0.886625358, 0.886726332, 0.887144874, 0.887375502, 0.887407612,  
0.888225108, 0.888308327, 0.888636372, 0.888663325, 0.888769328, 0.888852444,  
0.889141328, 0.889639703, 0.889708521, 0.889726353, 0.889739752, 0.889995749,  
0.890095088, 0.890097511, 0.890104916, 0.890499394, 0.890875797, 0.891145369,  
0.891343917, 0.891398089, 0.891471219, 0.891658061, 0.891701055, 0.891742494,  
0.892069502, 0.892083959, 0.892646995, 0.89276307, 0.892832715, 0.892915178,  
0.892944052, 0.893048905, 0.893225157, 0.89343297, 0.893471435, 0.894477747,  
0.894759429, 0.894777074, 0.895931416, 0.896052223, 0.896444345, 0.896584255,  
0.896792686, 0.897086591, 0.897198277, 0.897415056, 0.897489191, 0.897493458,  
0.897500092, 0.897523846, 0.897529607, 0.897544356, 0.897669803, 0.897716491,  
0.898154243, 0.898278625, 0.898278672, 0.898451884, 0.898492465, 0.898498792,  
0.898722176, 0.899562415, 0.899693942, 0.899698745, 0.899827836, 0.899955528,  
0.900170082, 0.900230224, 0.900231555, 0.900709775, 0.900816877, 0.900824698,  
0.901454467, 0.901459146, 0.9017719, 0.901923974, 0.901952409, 0.901991908,  
0.90205221, 0.902073926, 0.902219665, 0.902261482, 0.902315363, 0.90233874,  
0.902460367, 0.902465603, 0.902513877, 0.902748339, 0.902862282, 0.903020474,  
0.903059593, 0.903079379, 0.90309414, 0.903233579, 0.903241493, 0.90351695,

0.903578014, 0.903675005, 0.903707581, 0.903813708, 0.903873539, 0.903934354,  
0.903966238, 0.903973886, 0.90409771, 0.904179361, 0.904410629, 0.904440404,  
0.904470763, 0.904674487, 0.904775419, 0.905031579, 0.905060258, 0.905072347,  
0.905160295, 0.905172469, 0.905258916, 0.905326557, 0.90540523, 0.905563246,  
0.905595865, 0.90563274, 0.905973527, 0.906076934, 0.906208044, 0.906257466,  
0.906265304, 0.906312057, 0.906418672, 0.906441831, 0.906690674, 0.906844148,  
0.906894063, 0.907238159, 0.907363953, 0.907530714, 0.907542429, 0.907684381,  
0.907704244, 0.907762609, 0.90778397, 0.907785827, 0.907856518, 0.907902169,  
0.907957597, 0.907971892, 0.90798139, 0.908002065, 0.908300678, 0.908350771,  
0.90835477, 0.908436006, 0.908486346, 0.908501039, 0.908534332, 0.908572682,  
0.908577371, 0.908624112, 0.908687345, 0.908735384, 0.908800712, 0.908815159,  
0.908857963, 0.908880638, 0.908906198, 0.908995392, 0.908999558, 0.909066214,  
0.90912934, 0.909168956, 0.90928407, 0.909316297, 0.909351328, 0.909405116,  
0.909608934, 0.909651567, 0.909774743, 0.909804413, 0.909822707, 0.90983805,  
0.909874461, 0.909877664, 0.90991595, 0.910042597, 0.910061336, 0.910088085,  
0.910175485, 0.910302119, 0.910319425, 0.91035189, 0.910434333, 0.910507283,  
0.910544219, 0.910564024, 0.91056602, 0.910801612, 0.910821703, 0.910849074,  
0.910862818, 0.911009988, 0.911046891, 0.911063442, 0.911066552, 0.911134484,  
0.911146889, 0.911187269, 0.911245127, 0.911259219, 0.911273407, 0.911534382,  
0.911558263, 0.911584027, 0.911651679, 0.911680136, 0.911702, 0.911754861,  
0.911860741, 0.911881825, 0.911911104, 0.912057045, 0.912117939, 0.912120443,  
0.912236196, 0.912285856, 0.912315554, 0.912364277, 0.912379958, 0.912383327,  
0.912414083, 0.912500237, 0.912545418, 0.912560912, 0.91256809, 0.912605379,  
0.912720115, 0.912839183, 0.912850828, 0.912875611, 0.912877177, 0.912920592,  
0.91293647, 0.913017055, 0.913044034, 0.913110956, 0.913152609, 0.913154285,  
0.913189029, 0.91322302, 0.913308541, 0.913399077, 0.913453085, 0.913477402,  
0.913550138, 0.913606905, 0.913671617, 0.913680702, 0.913713276, 0.913723696,  
0.913759914, 0.913890843, 0.913909061, 0.914001406  
standar deviasi : 0.036585905504777014  
variansi : 0.001338528481604473  
range : 0.19408517999999997  
nilai minimum : 0.719916226  
nilai maksimum : 0.914001406  
kuartil-0 : 0.719916226  
kuartil-1 : 0.86367575275  
kuartil-2 : 0.8900454185  
kuartil-3 : 0.907577917  
kuartil-4 : 0.914001406  
IQR : 0.04390216425000004  
skewness : -1.6234718222806501  
kurtosis : 5.876188885335204

## 4.6 6. AreaBulatan

```
[8]: # Ambil kolom AreaBulatan
AreaBulatan = df["AreaBulatan"]
# Tulis statistik deskriptif dari kolom AreaBulatan
```

```
print("Statistik deskriptif kolom AreaBulatan")
printStatistic(AreaBulatan)
```

```
Statistik deskriptif kolom AreaBulatan
count                : 500
mean                 : 4937.048
median               : 4857.0
modus                : 3802, 4913
standar deviasi      : 1011.6962549701573
variansi             : 1023529.3123206415
range                : 5141
nilai minimum        : 2579
nilai maksimum       : 7720
quartil-0            : 2579.0
quartil-1            : 4170.25
quartil-2            : 4857.0
quartil-3            : 5654.25
quartil-4            : 7720.0
IQR                  : 1484.0
skewness              : 0.2575600053152032
kurtosis              : 2.582424426089691
```

## 4.7 7. Diameter

```
[9]: # Ambil kolom Diameter
Diameter = df["Diameter"]
# Tulis statistik deskriptif dari kolom Diameter
print("Statistik deskriptif kolom Diameter")
printStatistic(Diameter)
```

```
Statistik deskriptif kolom Diameter
count                : 500
mean                 : 77.77115780832001
median               : 77.64527658
modus                : 71.29356396, 78.83325579, 84.75622403, 88.00634154
standar deviasi      : 8.056867291849713
variansi             : 64.91311055847773
range                : 40.747172240000005
nilai minimum        : 56.66665803
nilai maksimum       : 97.41383027
quartil-0            : 56.66665803
quartil-1            : 71.7453075475
quartil-2            : 77.64527658
quartil-3            : 83.6485975675
quartil-4            : 97.41383027
IQR                  : 11.90329002
skewness              : 0.002724966865193717
kurtosis              : 2.526220942074553
```

## 4.8 8. KadarAir

```
[10]: # Ambil kolom KadarAir
KadarAir = df["KadarAir"]
# Tulis statistik deskriptif dari kolom KadarAir
print("Statistik deskriptif kolom KadarAir")
printStatistic(KadarAir)
```

```
Statistik deskriptif kolom KadarAir
count                : 500
mean                 : 0.6483716718979999
median               : 0.626116699
modus                : 0.735849057, 0.824404762
standar deviasi      : 0.09436709809379533
variansi             : 0.00890514920264399
range                : 0.46897193099999995
nilai minimum        : 0.409927152
nilai maksimum       : 0.878899083
quartil-0            : 0.409927152
quartil-1            : 0.57263245725
quartil-2            : 0.626116699
quartil-3            : 0.7266333445
quartil-4            : 0.878899083
IQR                  : 0.15400088724999994
skewness              : 0.49366131797330265
kurtosis              : 2.255080343612366
```

## 4.9 9. Keliling

```
[11]: # Ambil kolom Keliling
Keliling = df["Keliling"]
# Tulis statistik deskriptif dari kolom Keliling
print("Statistik deskriptif kolom Keliling")
printStatistic(Keliling)
```

```
Statistik deskriptif kolom Keliling
count                : 500
mean                 : 281.47972199999999
median               : 280.0455
modus                : 197.015, 200.587, 202.456, 207.325, 207.697, 208.317,
209.823, 210.012, 210.657, 211.667, 214.338, 214.44, 216.93, 218.773, 219.663,
221.295, 221.38, 222.373, 223.117, 224.485, 225.237, 226.049, 226.454, 226.793,
227.007, 227.562, 227.853, 227.906, 227.934, 228.007, 229.044, 229.787, 229.852,
230.332, 230.728, 230.804, 231.291, 232.122, 232.838, 232.94, 233.736, 234.047,
234.302, 234.781, 234.817, 235.385, 235.476, 235.534, 235.807, 235.956, 236.521,
236.767, 236.813, 237.412, 237.568, 237.593, 238.395, 238.547, 239.34, 239.356,
239.364, 240.017, 240.094, 240.295, 240.315, 240.44, 240.529, 240.546, 240.603,
```

240.915, 240.926, 241.24, 241.481, 241.542, 242.064, 242.117, 242.167, 242.294,  
242.407, 242.845, 243.201, 243.28, 243.849, 243.983, 244.123, 244.287, 244.49,  
245.108, 245.517, 245.889, 246.555, 247.032, 247.294, 247.69, 247.889, 248.428,  
249.213, 249.248, 249.61, 249.882, 250.098, 250.412, 250.912, 251.05, 251.165,  
251.355, 251.807, 252.331, 252.523, 252.837, 252.87, 253.068, 253.076, 253.449,  
253.493, 253.792, 254.662, 254.699, 254.771, 254.977, 255.102, 255.109, 255.508,  
255.689, 255.796, 255.912, 255.914, 256.199, 256.222, 256.509, 256.524, 256.647,  
256.711, 256.915, 257.06, 257.1, 257.268, 257.486, 257.607, 257.666, 258.503,  
259.184, 259.272, 259.409, 259.741, 259.808, 260.209, 260.346, 260.354, 260.462,  
260.478, 260.512, 260.732, 260.888, 260.959, 261.064, 261.089, 261.29, 261.405,  
261.474, 261.568, 261.701, 261.724, 261.858, 261.978, 262.097, 262.446, 262.743,  
263.0, 263.151, 263.299, 263.424, 263.639, 263.678, 263.752, 263.959, 264.24,  
264.28, 264.342, 264.418, 264.64, 264.644, 264.923, 264.951, 265.142, 265.151,  
265.167, 265.248, 265.364, 265.778, 266.221, 266.484, 266.682, 266.82, 266.923,  
267.193, 267.246, 267.759, 268.026, 268.297, 268.492, 268.59, 268.805, 268.84,  
268.932, 269.116, 269.153, 269.177, 269.387, 269.449, 269.51, 270.072, 270.638,  
270.823, 271.228, 271.738, 271.783, 272.761, 272.888, 273.085, 273.607, 273.621,  
273.942, 274.007, 274.231, 274.262, 274.318, 274.329, 274.396, 274.486, 274.946,  
275.362, 275.478, 275.818, 276.025, 276.786, 276.798, 276.911, 277.133, 277.216,  
277.531, 278.57, 278.571, 278.618, 278.623, 278.741, 279.019, 279.034, 279.606,  
279.955, 280.136, 280.448, 280.459, 280.5, 281.305, 281.35, 281.478, 281.839,  
282.031, 282.086, 282.244, 282.98, 283.186, 283.196, 283.33, 283.668, 283.809,  
283.873, 283.977, 284.145, 284.222, 284.343, 284.386, 284.75, 285.184, 285.254,  
285.494, 285.799, 286.116, 286.264, 286.377, 286.773, 286.844, 287.087, 287.436,  
287.462, 287.727, 287.874, 288.12, 288.273, 288.61, 288.779, 288.913, 289.009,  
289.043, 289.067, 289.5, 289.902, 289.962, 290.011, 290.047, 290.143, 291.224,  
291.302, 291.359, 291.454, 291.523, 291.54, 291.699, 292.196, 292.22, 292.518,  
292.569, 292.789, 292.865, 292.906, 293.36, 293.409, 293.548, 293.758, 293.922,  
294.219, 294.451, 294.789, 294.808, 294.988, 295.253, 295.612, 295.728, 296.2,  
296.211, 296.222, 297.081, 297.114, 297.14, 297.24, 297.276, 297.395, 297.503,  
298.048, 298.057, 298.992, 299.044, 299.083, 299.091, 299.67, 299.788, 299.803,  
299.924, 300.022, 300.597, 300.837, 300.885, 301.307, 301.902, 301.985, 302.11,  
302.131, 302.164, 302.542, 302.73, 302.928, 302.958, 303.047, 303.102, 303.285,  
303.999, 304.449, 305.036, 305.169, 305.326, 305.582, 305.628, 305.701, 305.872,  
306.634, 306.907, 306.914, 307.129, 307.304, 307.377, 307.543, 307.776, 307.947,  
307.965, 308.106, 308.355, 308.667, 309.111, 309.264, 309.712, 309.718, 309.826,  
310.522, 310.778, 310.785, 311.005, 311.235, 311.51, 311.647, 311.85, 311.868,  
311.871, 312.248, 312.386, 312.678, 312.835, 312.898, 313.011, 313.296, 313.372,  
314.233, 314.352, 314.656, 315.431, 315.483, 316.144, 316.194, 316.558, 316.683,  
316.69, 316.756, 316.987, 317.21, 317.226, 317.55, 317.671, 318.791, 318.86,  
318.914, 319.702, 319.883, 320.105, 320.388, 320.825, 320.902, 321.246, 321.354,  
321.441, 321.58, 321.778, 321.893, 322.073, 322.15, 322.198, 322.299, 322.3,  
322.853, 323.587, 323.807, 324.563, 326.626, 328.178, 328.799, 329.128, 329.747,  
330.709, 331.296, 331.417, 331.535, 331.893, 332.61, 333.722, 334.743, 336.166,  
336.803, 337.028, 338.191, 338.258, 338.782, 339.413, 340.02, 340.6, 340.726,  
341.702, 342.257, 342.99, 343.706, 343.917, 344.138, 346.308, 346.714, 347.364,  
348.085, 348.74, 352.718, 353.032, 353.285, 354.483, 357.414, 359.911, 362.021,  
365.062, 366.62, 367.844, 375.651, 390.125, 434.235, 448.305, 488.837

```

standar deviasi      : 37.33540171514401
variansi             : 1393.9322212311781
range                : 291.822
nilai minimum        : 197.015
nilai maksimum       : 488.837
kuartil-0            : 197.015
kuartil-1            : 255.883
kuartil-2            : 280.0455
kuartil-3            : 306.0625
kuartil-4            : 488.837
IQR                  : 50.179499999999999
skewness             : 0.7336269072005543
kurtosis             : 5.238045380548189

```

#### 4.10 10. Bulatan

```

[12]: # Ambil kolom Bulatan
      Bulatan = df["Bulatan"]
      # Tulis statistik deskriptif dari kolom Bulatan
      print("Statistik deskriptif kolom Bulatan")
      printStatistic(Bulatan)

```

Statistik deskriptif kolom Bulatan

```

count                : 500
mean                 : 0.7617374363080001
median               : 0.7612884394999999
modus                : 0.174590178, 0.261297389, 0.299297624, 0.589145961,
0.603806625, 0.61813153, 0.619331367, 0.622202152, 0.642939586, 0.655423795,
0.661848613, 0.671239887, 0.671414526, 0.67177503, 0.672018253, 0.673291547,
0.675018795, 0.676292881, 0.677200071, 0.679892535, 0.680194778, 0.685223012,
0.685880547, 0.687464035, 0.688159974, 0.688452216, 0.691149934, 0.694162454,
0.694463393, 0.695464012, 0.696051965, 0.696800862, 0.69743449, 0.698340924,
0.698558422, 0.700564861, 0.700632999, 0.704582713, 0.704652524, 0.704655156,
0.705696824, 0.707129779, 0.707728756, 0.708148034, 0.708372185, 0.708675046,
0.709805576, 0.711073136, 0.7112693, 0.711676727, 0.712019984, 0.712775871,
0.712803096, 0.713243729, 0.713441538, 0.713764078, 0.714924239, 0.715188526,
0.716227104, 0.716729844, 0.716912724, 0.716935365, 0.717492872, 0.717604447,
0.718366986, 0.718495828, 0.718778046, 0.718848061, 0.719381386, 0.719438986,
0.719483929, 0.719729966, 0.72004593, 0.720763075, 0.720945495, 0.721307267,
0.721905165, 0.722225121, 0.722225913, 0.72233116, 0.722573064, 0.722913536,
0.72303066, 0.723724085, 0.724040956, 0.724236695, 0.724437161, 0.724625404,
0.725654115, 0.726486094, 0.726768189, 0.726940396, 0.727187364, 0.727303504,
0.727618022, 0.727731578, 0.727872041, 0.72791653, 0.72813192, 0.728334216,
0.728371839, 0.72877966, 0.728838468, 0.728850092, 0.729070353, 0.729098899,
0.7291904, 0.729193361, 0.729245028, 0.729837764, 0.729843396, 0.729885004,
0.730130428, 0.730338017, 0.730424118, 0.730484673, 0.730528258, 0.730700321,
0.730784721, 0.730878349, 0.730952703, 0.731032195, 0.731302515, 0.731760731,
0.73190872, 0.732018064, 0.732158776, 0.733107749, 0.733198532, 0.733822792,

```

0.734584626, 0.734781582, 0.735078678, 0.73541536, 0.735723185, 0.735928796,  
0.736020303, 0.736410273, 0.736638788, 0.73669733, 0.736858714, 0.737621708,  
0.737915693, 0.738493716, 0.738745817, 0.73911166, 0.739239922, 0.739391972,  
0.739454838, 0.739489757, 0.739535505, 0.739701145, 0.739866625, 0.740104792,  
0.740328692, 0.740353291, 0.740358056, 0.740395924, 0.740608116, 0.740933276,  
0.741020574, 0.741521259, 0.74158786, 0.742026658, 0.742343348, 0.742366933,  
0.742402383, 0.743088467, 0.743899699, 0.743961322, 0.744169758, 0.744665063,  
0.744666469, 0.744958238, 0.74528593, 0.745372367, 0.74546102, 0.745835125,  
0.74588395, 0.746220249, 0.746240285, 0.746368785, 0.746530828, 0.747240192,  
0.74763833, 0.747917752, 0.748498263, 0.748551863, 0.748800888, 0.748807478,  
0.749109947, 0.749332104, 0.749377729, 0.749425424, 0.749637485, 0.749755204,  
0.749855551, 0.749903393, 0.750106052, 0.750124729, 0.750241913, 0.750407568,  
0.750728936, 0.750810515, 0.751010516, 0.751432462, 0.751462618, 0.751829344,  
0.751958363, 0.751993104, 0.752195573, 0.752337502, 0.752407832, 0.752786151,  
0.752894922, 0.753590762, 0.753627617, 0.753753835, 0.753836351, 0.753915777,  
0.754108849, 0.754232657, 0.754346393, 0.754745273, 0.755223412, 0.755649016,  
0.756274922, 0.756291127, 0.756676137, 0.756788434, 0.757257729, 0.757482389,  
0.757801052, 0.757811359, 0.758235653, 0.758298138, 0.758550524, 0.758673593,  
0.758791099, 0.758909935, 0.759156632, 0.759343932, 0.759513885, 0.75960905,  
0.760096593, 0.760508352, 0.760553426, 0.760729343, 0.761001621, 0.761173449,  
0.76140343, 0.761615131, 0.761764323, 0.761844354, 0.761919282, 0.762327135,  
0.762360758, 0.762366875, 0.762370689, 0.763007658, 0.763719957, 0.764005319,  
0.764034545, 0.764173999, 0.764377848, 0.764509615, 0.764572695, 0.764967217,  
0.765939166, 0.765946823, 0.766559782, 0.76667865, 0.7669605, 0.767184214,  
0.76744487, 0.767705186, 0.768109333, 0.768347011, 0.768376855, 0.768594199,  
0.768969555, 0.769410253, 0.769500016, 0.769854556, 0.769931108, 0.770173828,  
0.77020482, 0.770507543, 0.770521464, 0.770834311, 0.771055838, 0.771139527,  
0.771249536, 0.77160738, 0.772111842, 0.772243853, 0.772310913, 0.772799791,  
0.772926226, 0.773000377, 0.77306749, 0.773216435, 0.77354417, 0.773708713,  
0.774095281, 0.774227397, 0.774529044, 0.774655487, 0.775028673, 0.77584873,  
0.775904773, 0.776428464, 0.776656761, 0.777030274, 0.777070027, 0.77708742,  
0.777856643, 0.777877063, 0.778730598, 0.778849783, 0.779566092, 0.780316084,  
0.780418169, 0.780685775, 0.780850572, 0.781063565, 0.781359586, 0.78172962,  
0.782088589, 0.782091577, 0.782098362, 0.78220059, 0.782638217, 0.782646035,  
0.783326826, 0.784233206, 0.78490739, 0.784916415, 0.785459883, 0.786155688,  
0.786284485, 0.786601742, 0.78688525, 0.786886027, 0.787307713, 0.787310851,  
0.787784638, 0.788439188, 0.788440191, 0.788641635, 0.788992229, 0.7898489,  
0.790284652, 0.790389013, 0.790600715, 0.791474665, 0.791538634, 0.791576426,  
0.791709588, 0.791829553, 0.792412996, 0.79271154, 0.792889153, 0.793048786,  
0.793090693, 0.793107675, 0.793207209, 0.793430846, 0.793697097, 0.794065843,  
0.795189689, 0.795219134, 0.795386222, 0.795416754, 0.79633467, 0.796439869,  
0.797016671, 0.797464573, 0.798434479, 0.798439143, 0.798467606, 0.798748686,  
0.799155176, 0.799324168, 0.799740948, 0.800475639, 0.800638639, 0.800934592,  
0.801014362, 0.801637337, 0.80223953, 0.802261185, 0.802267072, 0.802603082,  
0.802984814, 0.803032283, 0.803149216, 0.803181878, 0.803461442, 0.803795109,  
0.804295624, 0.804545396, 0.804733164, 0.805111386, 0.805394671, 0.805439521,  
0.805564948, 0.805622906, 0.805732935, 0.806172585, 0.806779542, 0.806792259,  
0.807746273, 0.807792234, 0.808248354, 0.80846584, 0.809002546, 0.809504955,

```

0.80953786, 0.810257323, 0.811021355, 0.811599045, 0.812082213, 0.812242173,
0.812290673, 0.812695963, 0.812934875, 0.81359911, 0.814167476, 0.814449173,
0.814581787, 0.815019317, 0.815233866, 0.815258748, 0.815458208, 0.815718178,
0.816090663, 0.816500322, 0.81676783, 0.816931269, 0.817044617, 0.817635526,
0.819650226, 0.821763909, 0.822505549, 0.823987615, 0.824662357, 0.824880825,
0.82671961, 0.82721721, 0.827395697, 0.827930478, 0.82841931, 0.829487132,
0.829668616, 0.831320927, 0.831658201, 0.831797143, 0.833622792, 0.834805795,
0.835203461, 0.835810103, 0.836293963, 0.836830469, 0.83780329, 0.838935683,
0.839576062, 0.840438522, 0.841074214, 0.841438302, 0.841529145, 0.842805012,
0.843048377, 0.843860039, 0.844452833, 0.84511253, 0.846094886, 0.846535408,
0.847519424, 0.847593431, 0.848319358, 0.849296679, 0.84989101, 0.850158921,
0.850306436, 0.850763371, 0.856198377, 0.85847279, 0.863233761, 0.864092271,
0.867147782, 0.868433574, 0.868444056, 0.870203116, 0.870745605, 0.872416898,
0.874242929, 0.874743279, 0.891705551, 0.904748313
standar deviasi      : 0.06170246078673261
variansi             : 0.0038071936671382756
range                : 0.730158135
nilai minimum        : 0.174590178
nilai maksimum       : 0.904748313
kuartil-0            : 0.174590178
kuartil-1            : 0.731990728
kuartil-2            : 0.7612884394999999
kuartil-3            : 0.79636096975
kuartil-4            : 0.904748313
IQR                  : 0.06437024175000006
skewness              : -3.599236766361642
kurtosis              : 32.66420700471349

```

## 4.11 11. Ransum

```

[13]: # Ambil kolom Ransum
Ransum = df["Ransum"]
# Tulis statistik deskriptif dari kolom Ransum
print("Statistik deskriptif kolom Ransum")
printStatistic(Ransum)

```

```

Statistik deskriptif kolom Ransum
count      : 500
mean       : 2.150915331084
median     : 2.1935990365
modus      : 1.440795615, 1.453136582, 1.465950153, 1.48345605,
1.51000024, 1.519341968, 1.521727227, 1.540486631, 1.542057903, 1.547183245,
1.547535657, 1.550094178, 1.564652561, 1.580189539, 1.586608465, 1.587462159,
1.590134084, 1.618982021, 1.653079386, 1.65543321, 1.668112772, 1.675639128,
1.680039305, 1.683373724, 1.686592141, 1.688196232, 1.697250688, 1.699226473,
1.700643731, 1.706977459, 1.713914209, 1.724885488, 1.727032232, 1.736015342,
1.747523281, 1.75100193, 1.751274152, 1.760099997, 1.760210118, 1.763640445,
1.763657239, 1.778399476, 1.778595478, 1.780898286, 1.781414606, 1.782283909,

```



1.782779486, 1.788527458, 1.798166796, 1.799959097, 1.802645, 1.802893047,  
1.804012341, 1.805565702, 1.808158955, 1.812313365, 1.815781338, 1.817608766,  
1.826251811, 1.830091344, 1.833137796, 1.833740148, 1.837898676, 1.838250957,  
1.840286422, 1.840560518, 1.84134338, 1.846730754, 1.850139146, 1.851541036,  
1.853012623, 1.853650857, 1.8537631, 1.854013623, 1.855220886, 1.859079722,  
1.863828568, 1.875782413, 1.877694599, 1.878050455, 1.880645909, 1.883101662,  
1.884341011, 1.886439592, 1.88785243, 1.88870019, 1.888832212, 1.896722072,  
1.898185373, 1.901437269, 1.906046537, 1.90661938, 1.906817334, 1.907783994,  
1.910787598, 1.911224709, 1.91282581, 1.916971463, 1.918080969, 1.91909264,  
1.920943046, 1.92591221, 1.928716161, 1.930818309, 1.931058138, 1.935698552,  
1.938606424, 1.93890281, 1.946440108, 1.94688595, 1.950330173, 1.952970432,  
1.958254629, 1.958395654, 1.962802744, 1.96817911, 1.972737596, 1.973088027,  
1.975266097, 1.976679441, 1.976932348, 1.978690245, 1.978891282, 1.978953101,  
1.983874746, 1.983960139, 1.984260147, 1.986325767, 1.986499144, 1.986632846,  
1.995374027, 1.995664532, 1.996392202, 1.997707221, 1.9980862, 2.000380768,  
2.001146508, 2.001516019, 2.003259365, 2.005042746, 2.006531105, 2.007892171,  
2.008504397, 2.011060749, 2.011183955, 2.01270268, 2.012776022, 2.013836657,  
2.014703498, 2.014789001, 2.016956794, 2.018840896, 2.021275625, 2.023376728,  
2.027951335, 2.029096926, 2.029729291, 2.030404595, 2.032184299, 2.033637817,  
2.039549572, 2.041790482, 2.045159464, 2.047817202, 2.048368308, 2.049992068,  
2.050361589, 2.05259056, 2.056203409, 2.057247083, 2.059366454, 2.062514909,  
2.063055118, 2.063136708, 2.063481892, 2.064307142, 2.066397837, 2.067587589,  
2.0678626, 2.068723061, 2.069011284, 2.069290354, 2.070935285, 2.071131797,  
2.071210372, 2.071483926, 2.073422024, 2.074970421, 2.075068371, 2.078458874,  
2.078462475, 2.078660032, 2.080344263, 2.08232503, 2.086528594, 2.087475392,  
2.090898855, 2.093760238, 2.097097749, 2.099186519, 2.099847143, 2.100274369,  
2.100666294, 2.102246978, 2.104916927, 2.104926204, 2.104945153, 2.105707298,  
2.105843517, 2.107114975, 2.107740189, 2.109368773, 2.110508466, 2.113900842,  
2.11410263, 2.118581367, 2.121543339, 2.122124407, 2.122443312, 2.123595479,  
2.124837558, 2.126628263, 2.128536971, 2.131495771, 2.132348203, 2.133619661,  
2.133648387, 2.136813669, 2.143551574, 2.149283251, 2.1525715, 2.153467052,  
2.156809452, 2.162216624, 2.163122241, 2.166889345, 2.168974314, 2.169265121,  
2.176712075, 2.177474846, 2.180490204, 2.18073856, 2.181716211, 2.182483784,  
2.185158479, 2.189798059, 2.190441243, 2.190608013, 2.190733345, 2.193132409,  
2.194065664, 2.194088445, 2.194158073, 2.19787742, 2.201445593, 2.204012677,  
2.205909669, 2.206428166, 2.207128755, 2.208922, 2.209335313, 2.209733911,  
2.212887655, 2.213027428, 2.218493122, 2.219625409, 2.220305672, 2.221112029,  
2.221394586, 2.222421679, 2.224151651, 2.226197042, 2.226576302, 2.236573722,  
2.239398401, 2.239575732, 2.251276639, 2.252512709, 2.256540017, 2.257982635,  
2.260137337, 2.263187001, 2.264349408, 2.266611124, 2.267386283, 2.267430923,  
2.267500333, 2.267748924, 2.267809234, 2.267963643, 2.269278355, 2.269768294,  
2.274378701, 2.275694235, 2.275694736, 2.277530842, 2.277961715, 2.278028913,  
2.280405651, 2.289417954, 2.290839165, 2.290891123, 2.292288889, 2.293674225,  
2.29600804, 2.296663622, 2.296678136, 2.301912694, 2.303090322, 2.303176396,  
2.310141398, 2.3101934, 2.313677987, 2.315378484, 2.315696898, 2.316139432,  
2.316815562, 2.317059217, 2.318696496, 2.31916698, 2.319773635, 2.32003701,  
2.321408831, 2.321467942, 2.322013197, 2.324667319, 2.325960705, 2.327760214,  
2.328205903, 2.328431431, 2.328599738, 2.330191507, 2.330281955, 2.333437182,

2.334138503, 2.335253858, 2.335628859, 2.33685187, 2.33754228, 2.338244708,  
2.338613257, 2.338701684, 2.340134912, 2.341081542, 2.343769511, 2.344116298,  
2.344470053, 2.346848427, 2.348029651, 2.35103612, 2.35137349, 2.35151575,  
2.352551509, 2.352695003, 2.353714728, 2.354513601, 2.355443876, 2.357315927,  
2.357702973, 2.358140757, 2.362199074, 2.363434967, 2.365004953, 2.365597637,  
2.365691672, 2.366252847, 2.367534178, 2.367812796, 2.370813286, 2.372669947,  
2.373274809, 2.377458021, 2.378993238, 2.381033342, 2.381176872, 2.382918257,  
2.383162254, 2.383879665, 2.384142396, 2.384165244, 2.385035427, 2.385597924,  
2.38628145, 2.386457846, 2.386575063, 2.386830293, 2.390526352, 2.391148176,  
2.391197843, 2.392207428, 2.392833732, 2.393016634, 2.393431242, 2.393909118,  
2.393967558, 2.394550452, 2.395339723, 2.395939907, 2.396756873, 2.396937668,  
2.397473566, 2.39775761, 2.398077931, 2.399196775, 2.399249066, 2.40008635,  
2.400880158, 2.401378756, 2.402829454, 2.403236086, 2.403678351, 2.404357946,  
2.406938681, 2.407479634, 2.409044724, 2.409422211, 2.40965505, 2.409850393,  
2.410314159, 2.410354967, 2.41084296, 2.412459428, 2.412698905, 2.413040879,  
2.414159316, 2.415782774, 2.416004902, 2.416421796, 2.417481504, 2.418420435,  
2.418896288, 2.41915155, 2.419177285, 2.422220627, 2.422480727, 2.422835222,  
2.42301328, 2.424922642, 2.425402178, 2.425617344, 2.425657774, 2.426541607,  
2.426703106, 2.427229072, 2.427983334, 2.428167151, 2.428352271, 2.43176546,  
2.432078558, 2.432416486, 2.43330455, 2.433678416, 2.433965791, 2.434661025,  
2.436055477, 2.43633347, 2.436719665, 2.438647615, 2.439453507, 2.439486662,  
2.441020983, 2.441680186, 2.442074669, 2.442722324, 2.442930883, 2.442975699,  
2.44338494, 2.44453248, 2.445134964, 2.445341696, 2.44543748, 2.445935282,  
2.447469031, 2.449063942, 2.449220104, 2.449552554, 2.449573567, 2.450156332,  
2.450369573, 2.451452735, 2.451815722, 2.452716836, 2.453278224, 2.453300825,  
2.453769434, 2.454228167, 2.455383527, 2.456608555, 2.457340248, 2.45766993,  
2.458656883, 2.45942803, 2.460308053, 2.460431687, 2.46087511, 2.461017015,  
2.461510443, 2.463296836, 2.463545729, 2.464808581

standar deviasi : 0.2497668892706582  
variansi : 0.06238349897594124  
range : 1.0240129660000001  
nilai minimum : 1.440795615  
nilai maksimum : 2.464808581  
kuartil-0 : 1.440795615  
kuartil-1 : 1.98393879075  
kuartil-2 : 2.1935990365  
kuartil-3 : 2.38161221825  
kuartil-4 : 2.464808581  
IQR : 0.3976734275  
skewness : -0.6581880925333655  
kurtosis : 2.5636427316999044

## 4.12 12. Kelas

Kelas tidak perlu ditulis deskripsi statistiknya karena kelas merepresentasikan kelas dari suatu gandum dan tidak bisa dihitung statistik deskriptifnya.

## 5 Section 2 - Data Visualization

Membuat Visualisasi plot distribusi, dalam bentuk histogram dan boxplot untuk setiap kolom numerik. Berikan uraian penjelasan kondisi setiap kolom berdasarkan kedua plot tersebut.

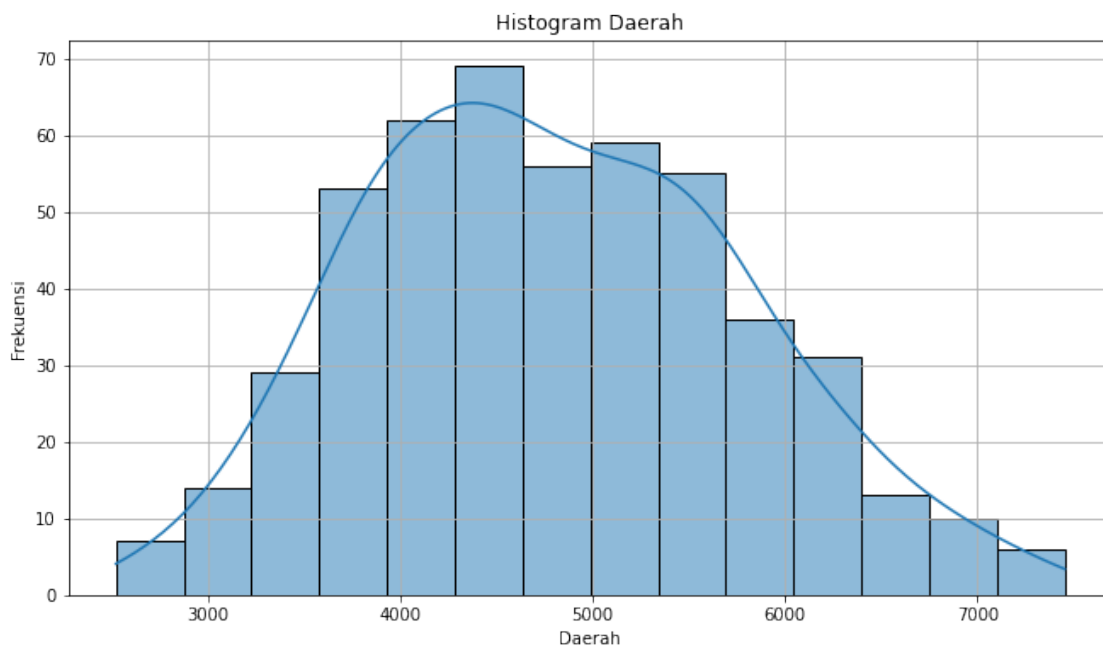
### 5.1 1. id

id hanyalah pembeda dari setiap row, sehingga tidak perlu dilakukan visualisasi terhadap kolom id sekalipun kolom ini numerik

### 5.2 2. Daerah

```
[14]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistDaerah =sns.histplot(Daerah, kde=True)
HistDaerah.set_title("Histogram Daerah")
HistDaerah.set_ylabel("Frekuensi")
```

```
[14]: Text(0, 0.5, 'Frekuensi')
```

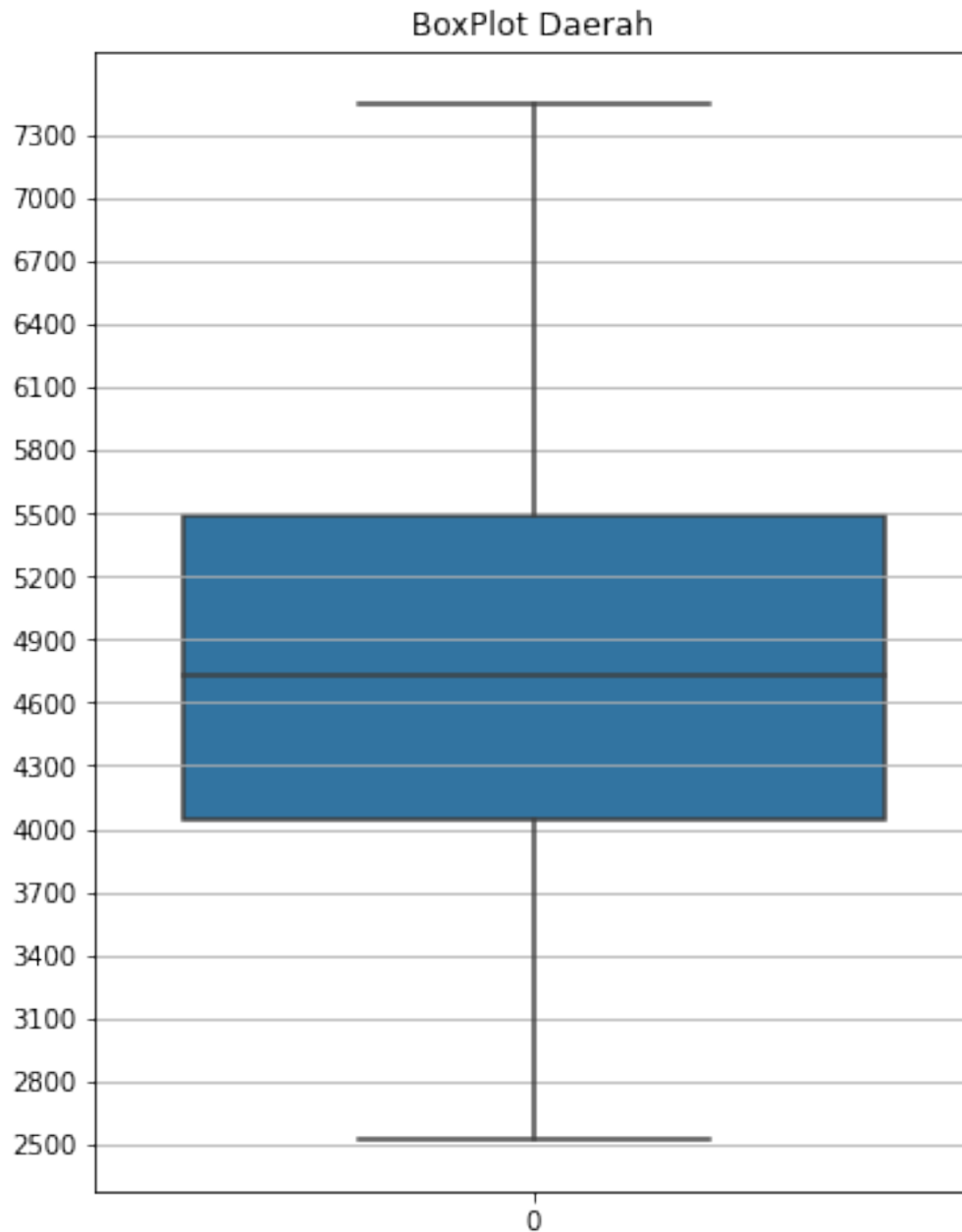


Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom Daerah terdistribusi secara simetris sehingga bersifat No Skew (Normal) dan berbentuk tidak runcing dan tidak landai sehingga bersifat Mesokurtis.

```
[15]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
```

```
plt.grid()
# Penggambaran Boxplot
BoxPlotDaerah= sns.boxplot(data=Daerah)
BoxPlotDaerah.set_title("BoxPlot Daerah")
plt.yticks(np.arange(int(round(min(Daerah),-2)), max(Daerah), 300))
```

```
[15]: ([<matplotlib.axis.YTick at 0x21c20f186a0>,
<matplotlib.axis.YTick at 0x21c20f10ee0>,
<matplotlib.axis.YTick at 0x21c20f102b0>,
<matplotlib.axis.YTick at 0x21c20d775b0>,
<matplotlib.axis.YTick at 0x21c20d77d00>,
<matplotlib.axis.YTick at 0x21c20d7e490>,
<matplotlib.axis.YTick at 0x21c20d77ca0>,
<matplotlib.axis.YTick at 0x21c20d7e460>,
<matplotlib.axis.YTick at 0x21c20d831c0>,
<matplotlib.axis.YTick at 0x21c20d83910>,
<matplotlib.axis.YTick at 0x21c20d8a160>,
<matplotlib.axis.YTick at 0x21c20d8a7f0>,
<matplotlib.axis.YTick at 0x21c20d8af40>,
<matplotlib.axis.YTick at 0x21c20d83b20>,
<matplotlib.axis.YTick at 0x21c20d77760>,
<matplotlib.axis.YTick at 0x21c20d945b0>,
<matplotlib.axis.YTick at 0x21c20d94d00>],
[Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, '')[19]])
```

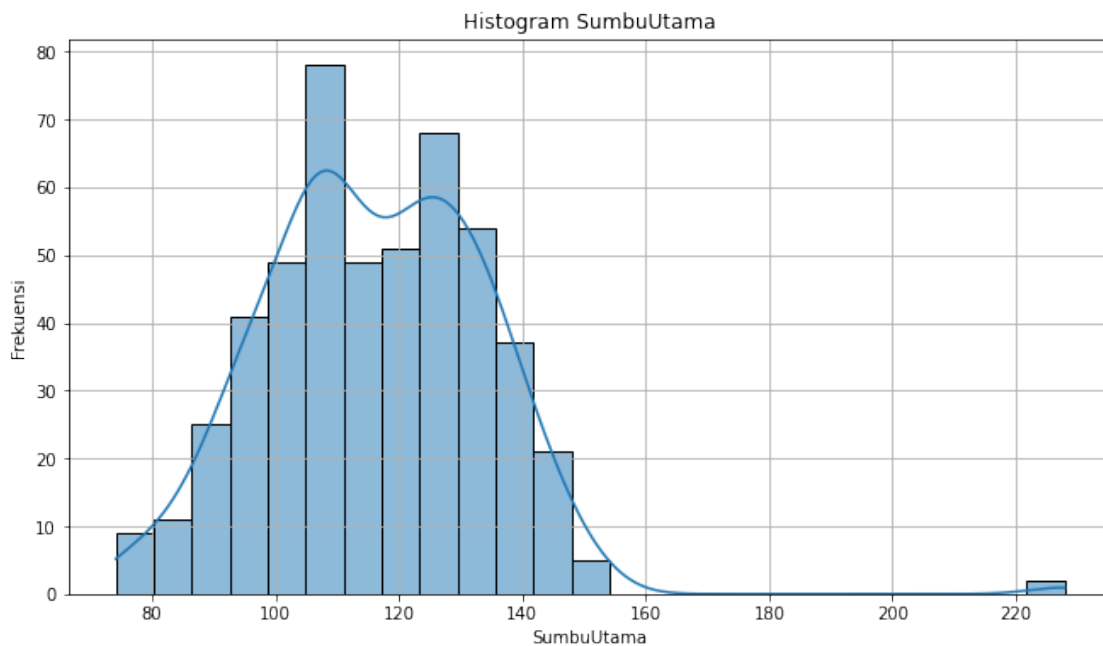


Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom Daerah sekitar 2500, nilai Q1 sekitar 4000, nilai Q2 atau median sekitar 4700, nilai Q3 sekitar 5500, dan nilai maksimal sekitar 7400. Boxplot ini juga menunjukkan bahwa kolom Daerah tidak memiliki pencilan.

### 5.3 3. SumbuUtama

```
[16]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistSumbuUtama =sns.histplot(SumbuUtama, kde=True)
HistSumbuUtama.set_title("Histogram SumbuUtama")
HistSumbuUtama.set_ylabel("Frekuensi")
```

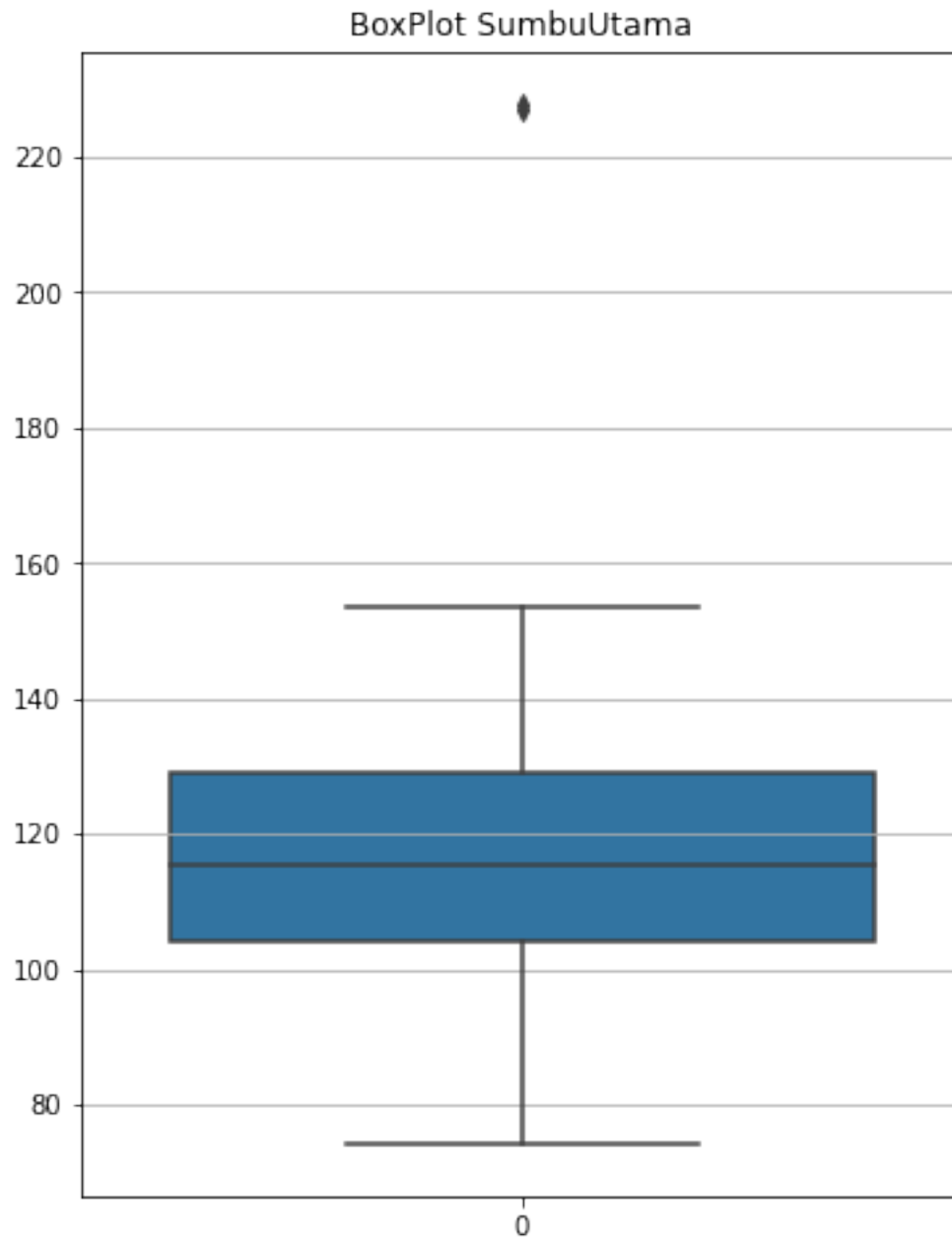
```
[16]: Text(0, 0.5, 'Frekuensi')
```



Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom SumbuUtama terdistribusi secara condong ke kanan sehingga bersifat Positively Skewed dan berbentuk runcing sehingga bersifat Leptokurtis.

```
[17]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot
BoxPlotSumbuUtama= sns.boxplot(data=SumbuUtama)
BoxPlotSumbuUtama.set_title("BoxPlot SumbuUtama")
```

```
[17]: Text(0.5, 1.0, 'BoxPlot SumbuUtama')
```

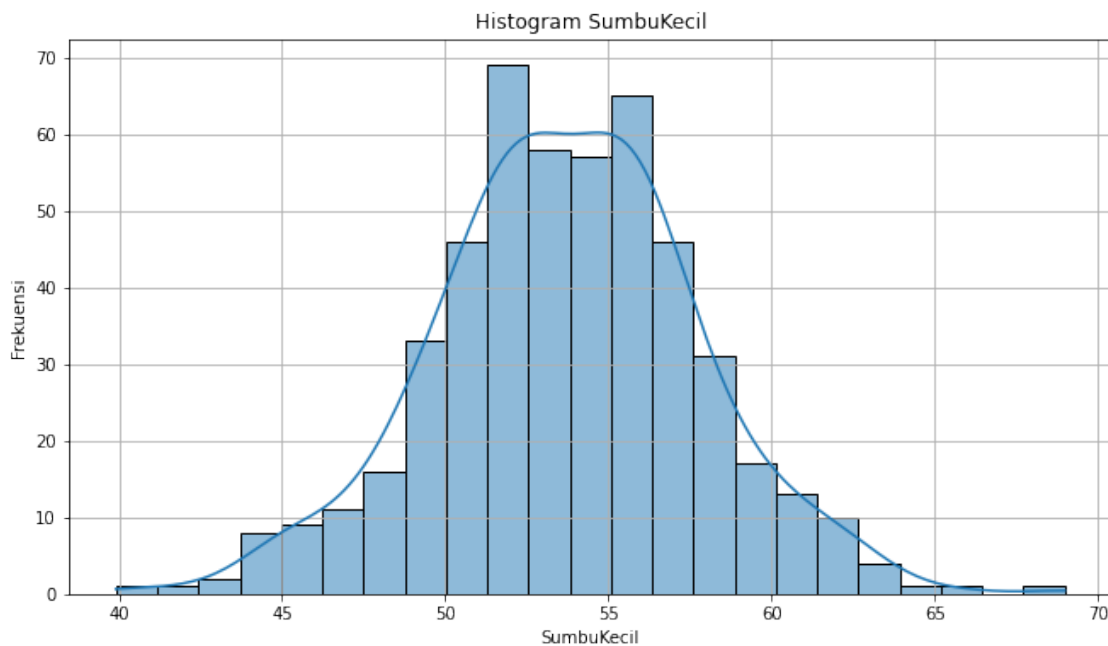


Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom SumbuUtama sekitar 75, nilai Q1 sekitar 105, nilai Q2 atau median sekitar 115, nilai Q3 sekitar 130, dan nilai maksimal sekitar 230. Boxplot ini juga menunjukan bahwa kolom SumbuUtama memiliki pencilan atas dengan nilai di sekitar 230.

## 5.4 4. SumbuKecil

```
[18]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistSumbuKecil =sns.histplot(SumbuKecil, kde=True)
HistSumbuKecil.set_title("Histogram SumbuKecil")
HistSumbuKecil.set_ylabel("Frekuensi")
```

```
[18]: Text(0, 0.5, 'Frekuensi')
```

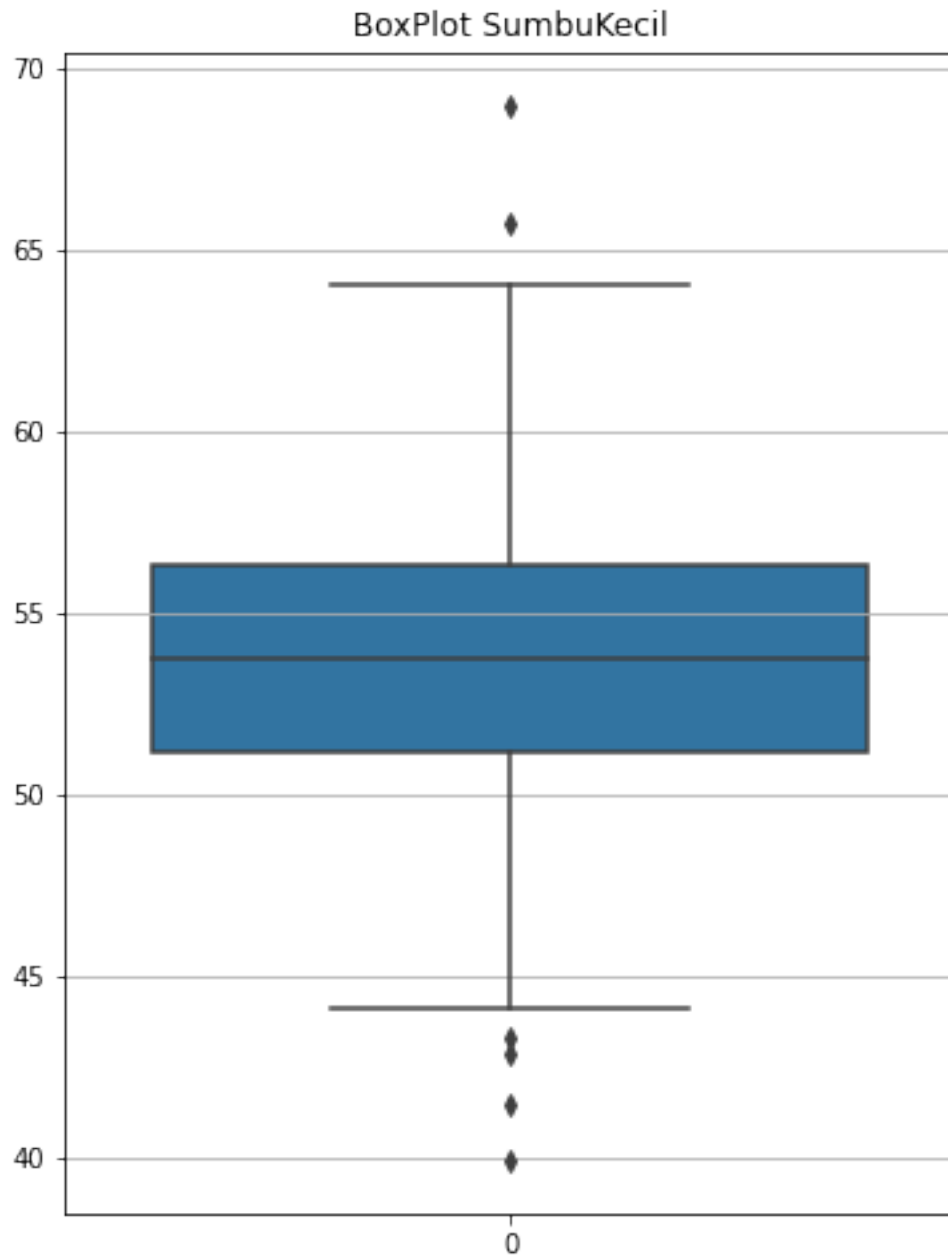


Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom SumbuKecil terdistribusi secara simetris sehingga bersifat No Skew (Normal) dan berbentuk tidak runcing dan tidak landai sehingga bersifat Mesokurtis.

```
[19]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot
BoxPlotSumbuKecil= sns.boxplot(data=SumbuKecil)
BoxPlotSumbuKecil.set_title("BoxPlot SumbuKecil")
```

```
[19]: Text(0.5, 1.0, 'BoxPlot SumbuKecil')
```



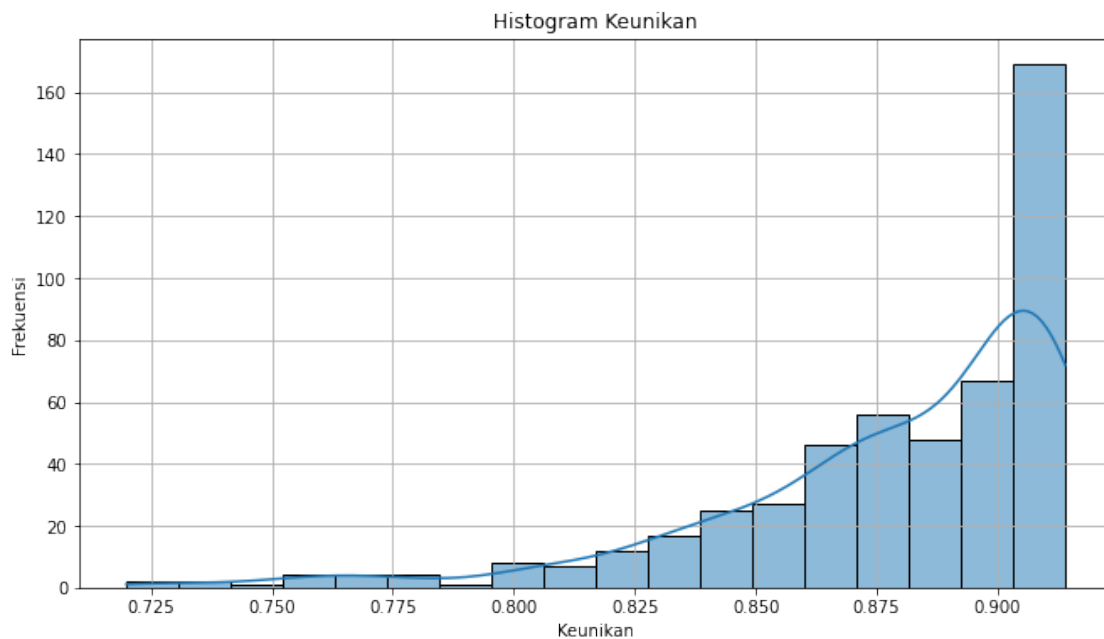


Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom SumbuKecil sekitar 40, nilai Q1 sekitar 51, nilai Q2 atau median sekitar 54, nilai Q3 sekitar 56, dan nilai maksimal sekitar 69. Boxplot ini juga menunjukkan bahwa kolom SumbuKecil memiliki beberapa pencilan yaitu 4 pencilan bawah dengan nilai di sekitar 40-44 dan 2 pencilan atas dengan nilai di sekitar 66-69.

## 5.5 5. Keunikan

```
[20]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistKeunikan = sns.histplot(Keunikan, kde=True)
HistKeunikan.set_title("Histogram Keunikan")
HistKeunikan.set_ylabel("Frekuensi")
```

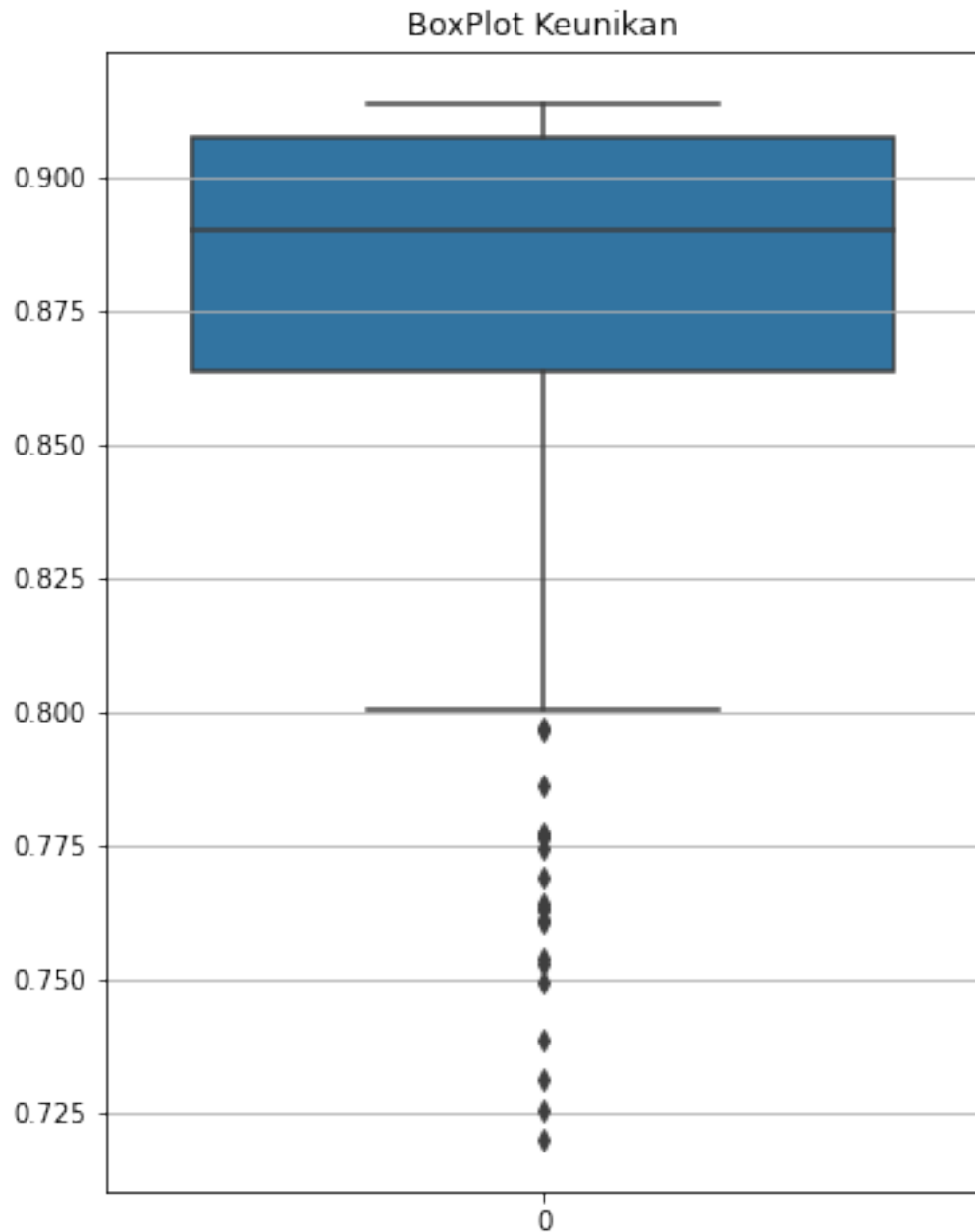
```
[20]: Text(0, 0.5, 'Frekuensi')
```



Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom Keunikan terdistribusi secara condong ke kiri sehingga bersifat Negatively Skewed dan berbentuk runcing sehingga bersifat Leptokurtis.

```
[21]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot
BoxPlotKeunikan= sns.boxplot(data=Keunikan)
BoxPlotKeunikan.set_title("BoxPlot Keunikan")
```

```
[21]: Text(0.5, 1.0, 'BoxPlot Keunikan')
```

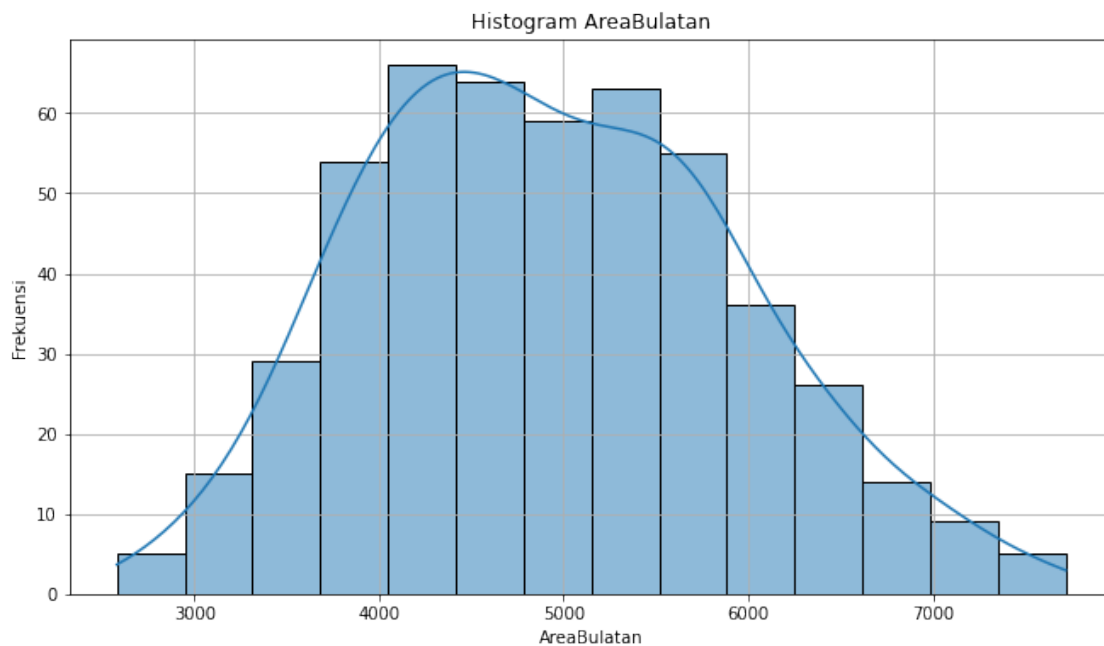


Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom Keunikan sekitar 0.72, nilai Q1 sekitar 0.865, nilai Q2 atau median sekitar 0.890, nilai Q3 sekitar 0.910, dan nilai maksimal sekitar 0.915. Boxplot ini juga menunjukan bahwa kolom Keunikan memiliki beberapa pencilan bawah dengan nilai dibawah 0.8.

## 5.6 6. AreaBulatan

```
[22]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistAreaBulatan = sns.histplot(AreaBulatan, kde=True)
HistAreaBulatan.set_title("Histogram AreaBulatan")
HistAreaBulatan.set_ylabel("Frekuensi")
```

```
[22]: Text(0, 0.5, 'Frekuensi')
```



Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom AreaBulatan terdistribusi secara simetris sehingga bersifat No Skew (Normal) dan berbentuk tidak runcing dan tidak landai sehingga bersifat Mesokurtis.

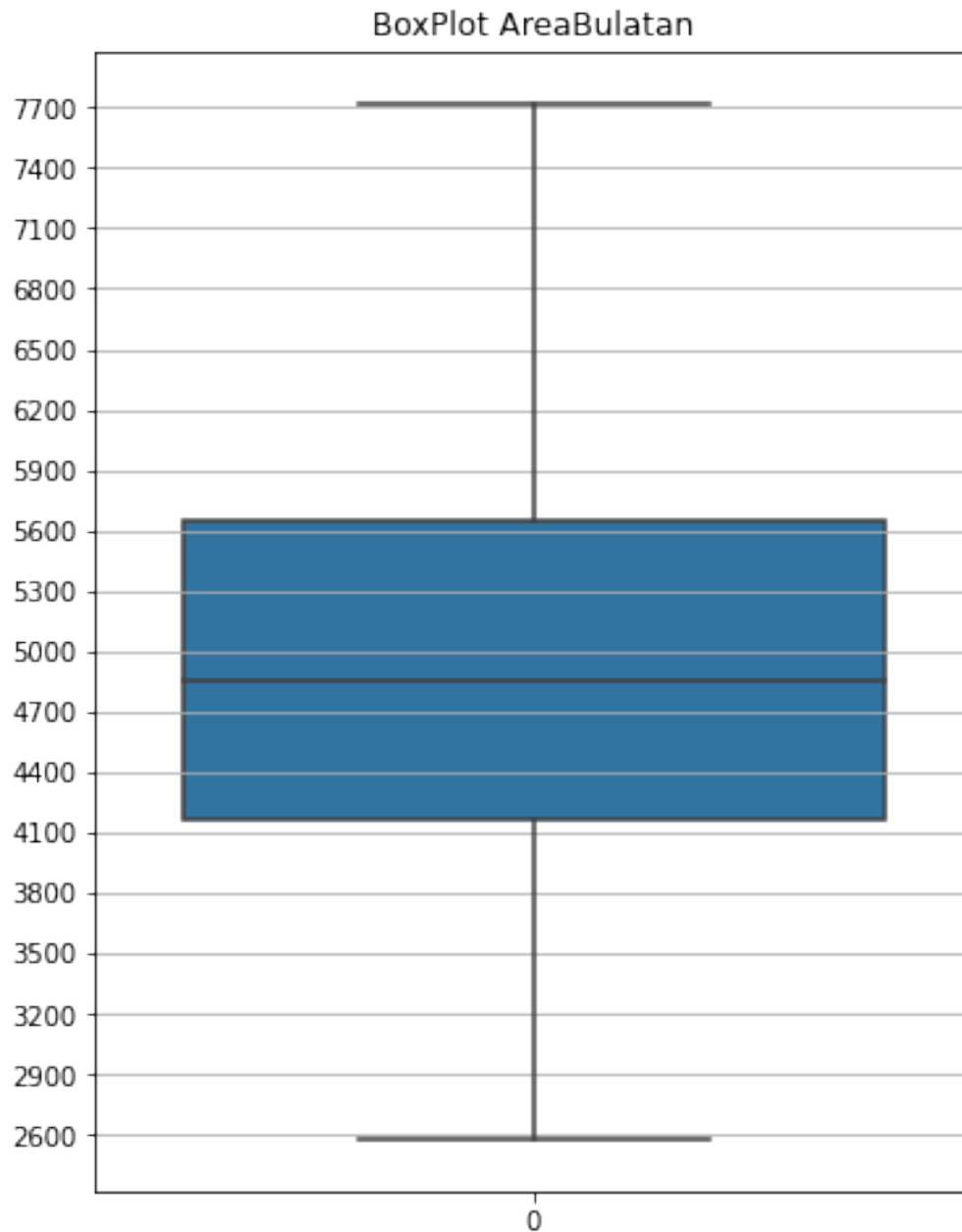
```
[23]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot
BoxPlotAreaBulatan = sns.boxplot(data=AreaBulatan)
BoxPlotAreaBulatan.set_title("BoxPlot AreaBulatan")
plt.yticks(np.arange(int(round(min(AreaBulatan),-2)), max(AreaBulatan), 300))
```

```
[23]: ([<matplotlib.axis.YTick at 0x21c21231b20>,
<matplotlib.axis.YTick at 0x21c212313a0>,
<matplotlib.axis.YTick at 0x21c2122d280>,
<matplotlib.axis.YTick at 0x21c21268eb0>,
<matplotlib.axis.YTick at 0x21c21271640>,
```

```

<matplotlib.axis.YTick at 0x21c21271d90>,
<matplotlib.axis.YTick at 0x21c21277520>,
<matplotlib.axis.YTick at 0x21c21277c70>,
<matplotlib.axis.YTick at 0x21c21277e80>,
<matplotlib.axis.YTick at 0x21c21271d60>,
<matplotlib.axis.YTick at 0x21c2127f370>,
<matplotlib.axis.YTick at 0x21c2127faf0>,
<matplotlib.axis.YTick at 0x21c21286280>,
<matplotlib.axis.YTick at 0x21c212869d0>,
<matplotlib.axis.YTick at 0x21c21287160>,
<matplotlib.axis.YTick at 0x21c21286670>,
<matplotlib.axis.YTick at 0x21c212688e0>,
<matplotlib.axis.YTick at 0x21c21287670>],
[Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, ''),
Text(0, 0, '')]

```

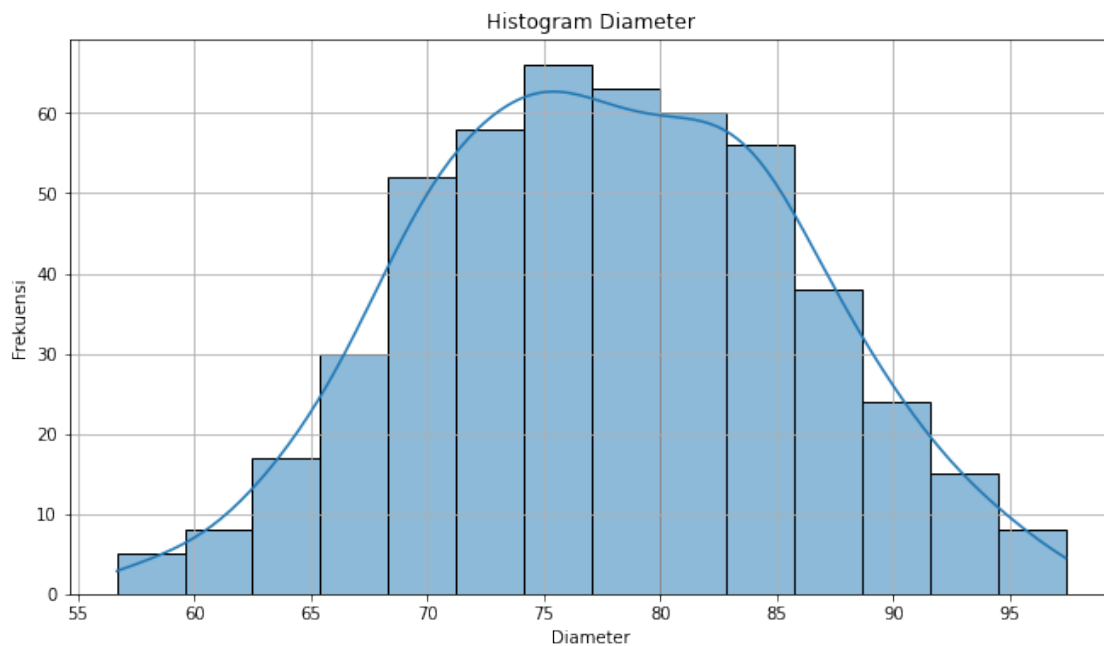


Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom AreaBulatan sekitar 2600, nilai Q1 sekitar 4200, nilai Q2 atau median sekitar 4900, nilai Q3 sekitar 5600, dan nilai maksimal sekitar 7700. Boxplot ini juga menunjukan bahwa kolom AreaBulatan tidak memiliki pencilan.

## 5.7 7. Diameter

```
[24]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistDiameter = sns.histplot(Diameter, kde=True)
HistDiameter.set_title("Histogram Diameter")
HistDiameter.set_ylabel("Frekuensi")
```

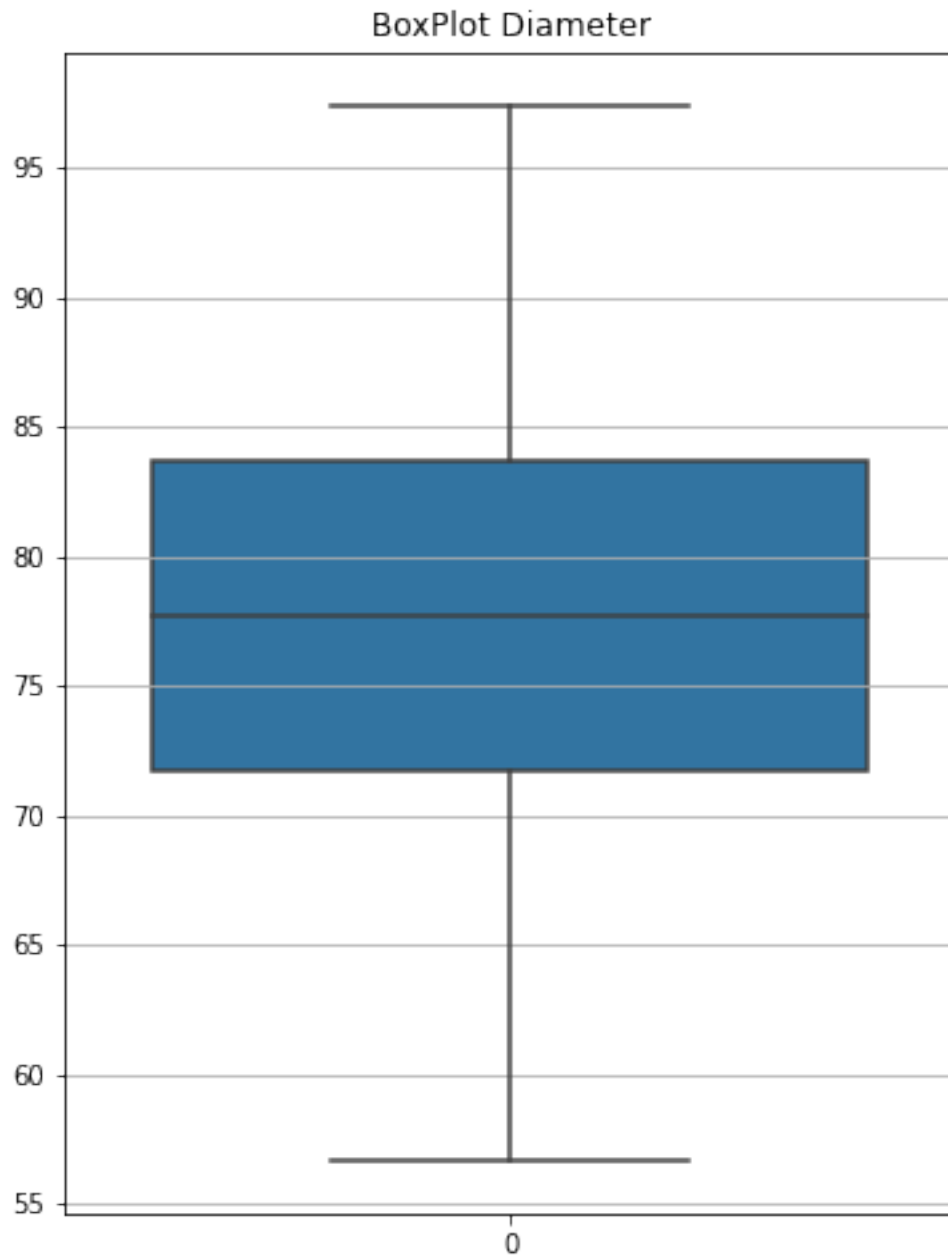
```
[24]: Text(0, 0.5, 'Frekuensi')
```



Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom Diameter terdistribusi secara simetris sehingga bersifat No Skew (Normal) dan berbentuk tidak runcing dan tidak landai sehingga bersifat Mesokurtis.

```
[25]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot
BoxPlotDiameter= sns.boxplot(data=Diameter)
BoxPlotDiameter.set_title("BoxPlot Diameter")
```

```
[25]: Text(0.5, 1.0, 'BoxPlot Diameter')
```



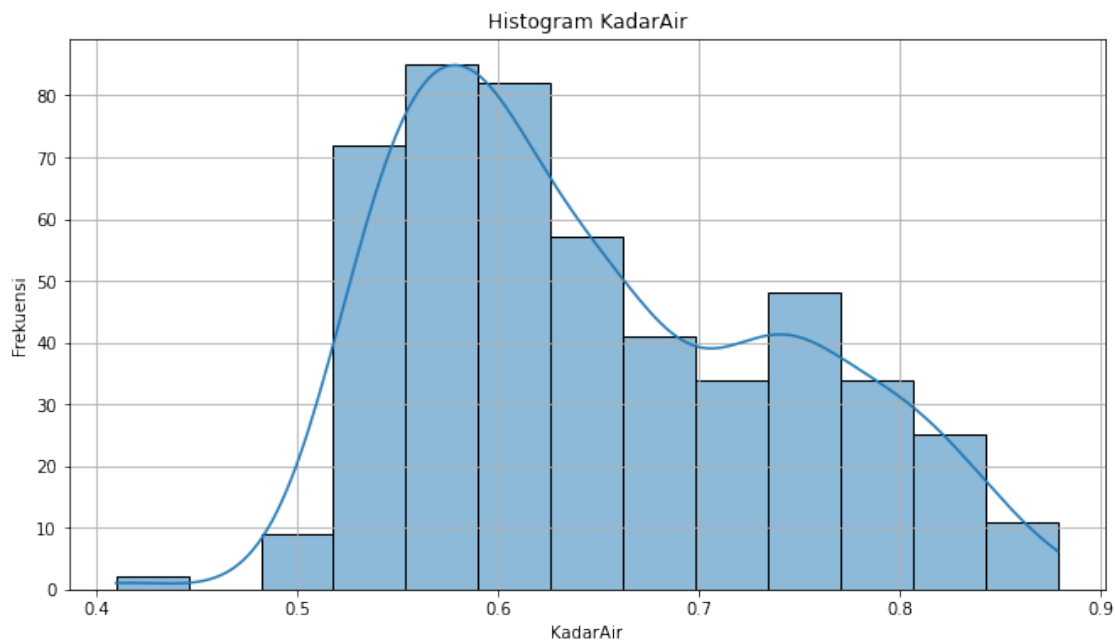
Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom Diameter sekitar 57, nilai Q1 sekitar 72, nilai Q2 atau median sekitar 78, nilai Q3 sekitar 84, dan nilai maksimal sekitar 97. Boxplot ini juga menunjukan bahwa kolom Diameter tidak memiliki pencilan.



## 5.8 8. Kadar Air

```
[26]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistKadarAir =sns.histplot(KadarAir, kde=True)
HistKadarAir.set_title("Histogram KadarAir")
HistKadarAir.set_ylabel("Frekuensi")
```

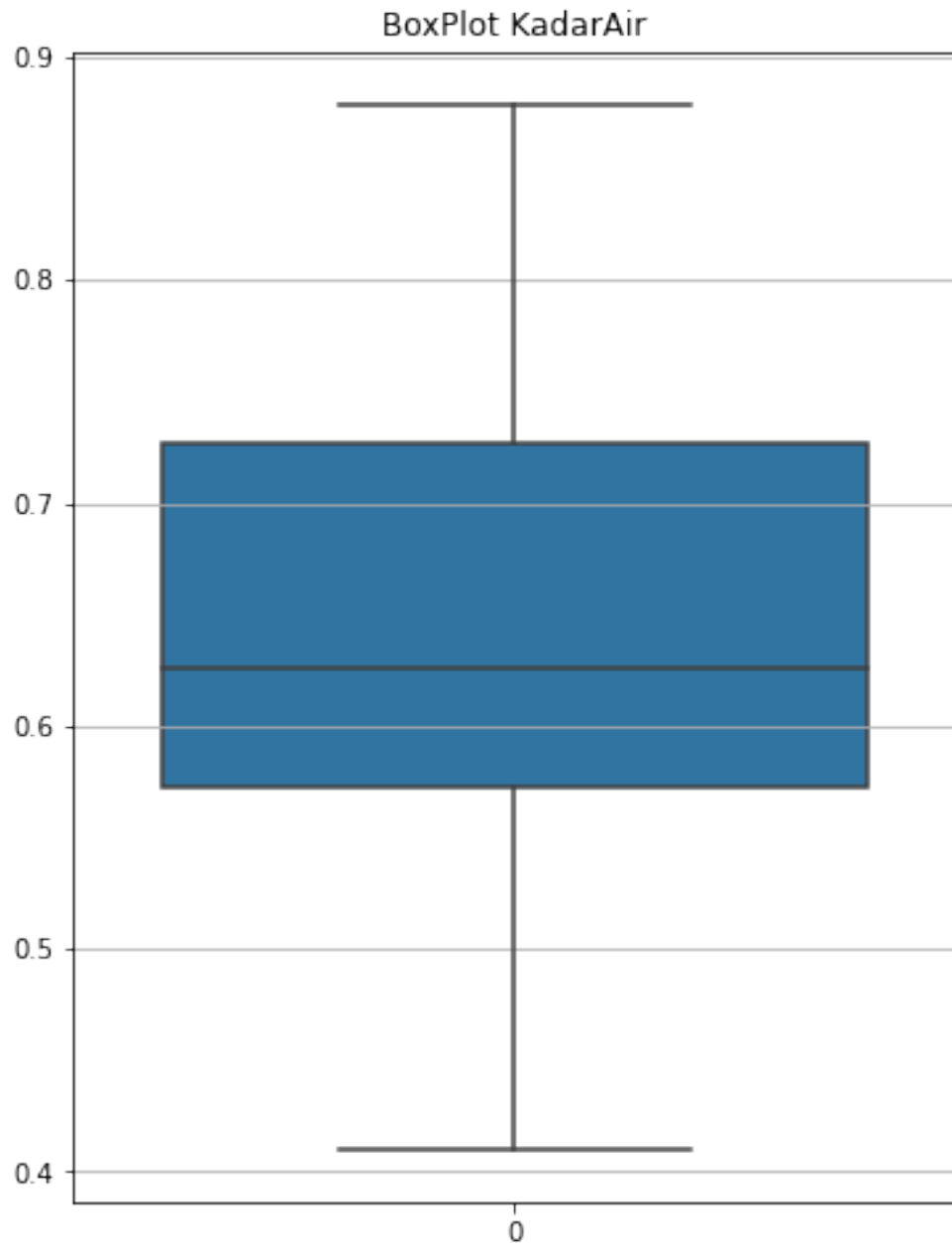
```
[26]: Text(0, 0.5, 'Frekuensi')
```



Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom KadarAir terdistribusi secara simetris sehingga bersifat No Skew (Normal) dan berbentuk landai sehingga bersifat Platykurtis.

```
[27]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot
BoxPlotKadarAir= sns.boxplot(data=KadarAir)
BoxPlotKadarAir.set_title("BoxPlot KadarAir")
```

```
[27]: Text(0.5, 1.0, 'BoxPlot KadarAir')
```

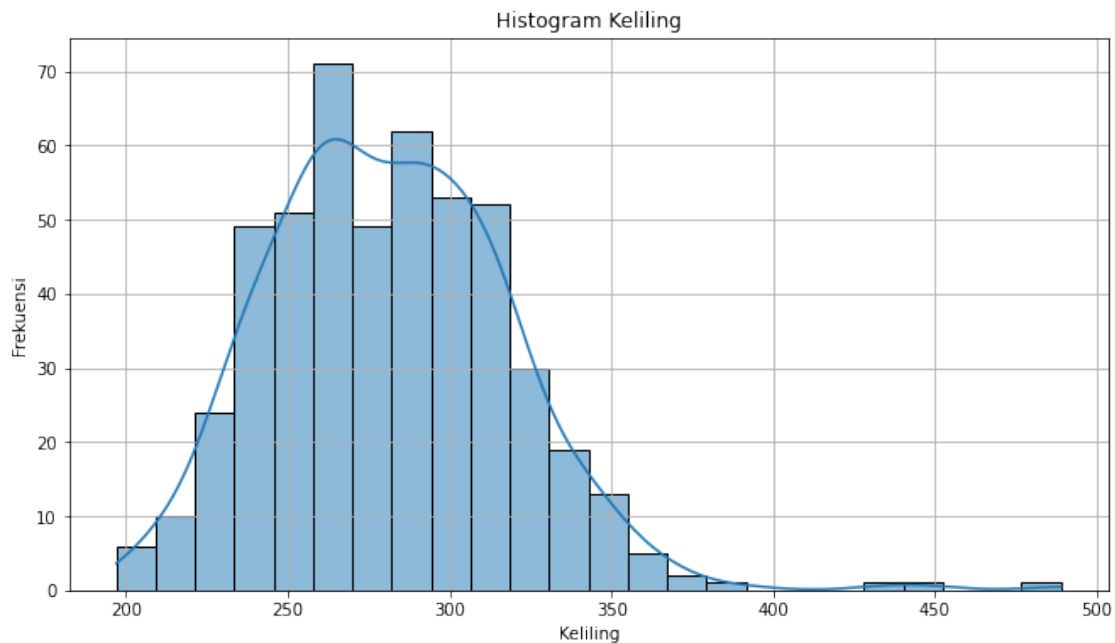


Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom KadarAir sekitar 0.41, nilai Q1 sekitar 0.58, nilai Q2 atau median sekitar 0.63, nilai Q3 sekitar 0.73, dan nilai maksimal sekitar 0.88. Boxplot ini juga menunjukan bahwa kolom KadarAir tidak memiliki pencilan.

## 5.9 9. Keliling

```
[28]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistKeliling =sns.histplot(Keliling, kde=True)
HistKeliling.set_title("Histogram Keliling")
HistKeliling.set_ylabel("Frekuensi")
```

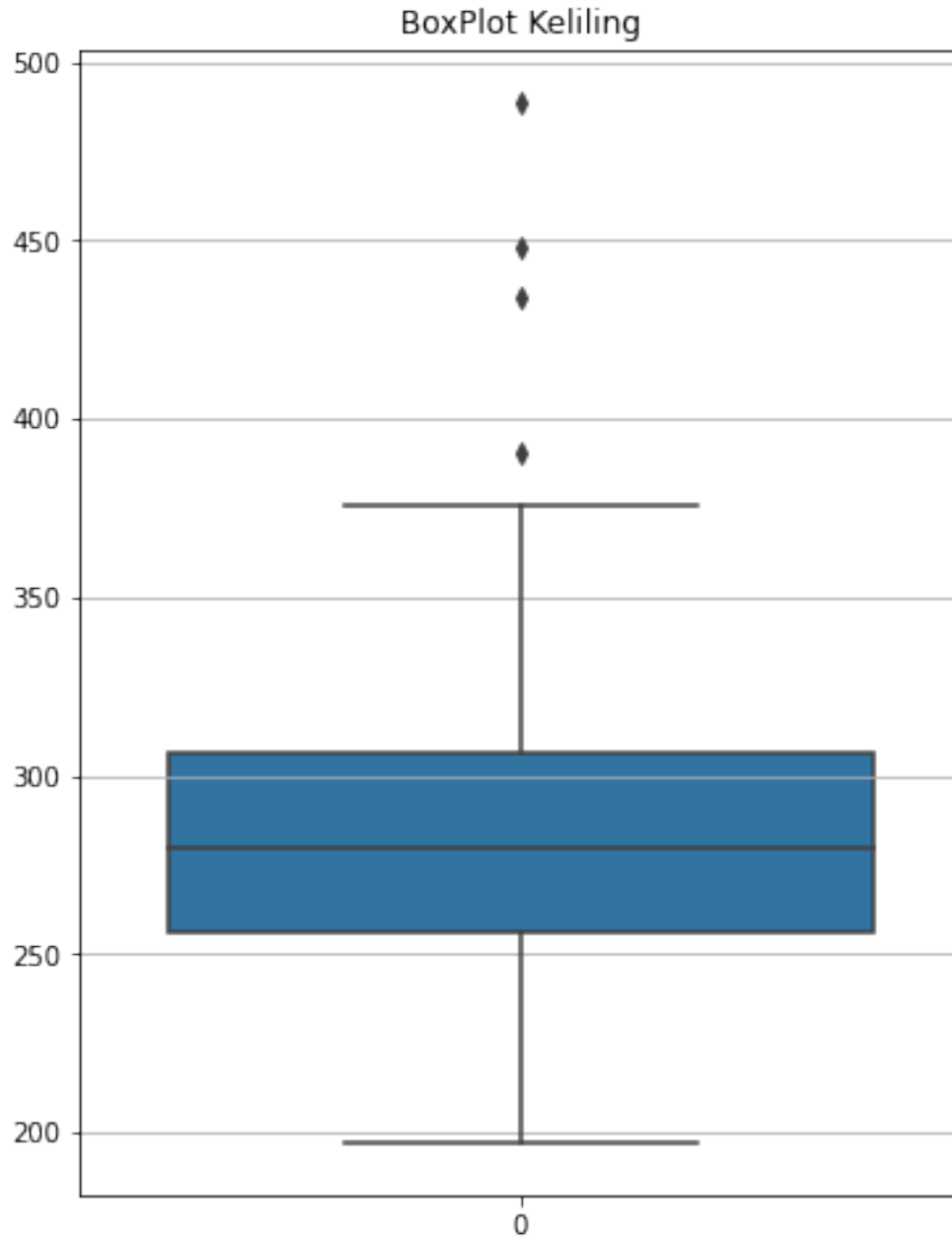
```
[28]: Text(0, 0.5, 'Frekuensi')
```



Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom Keliling terdistribusi secara condong ke kanan sehingga bersifat Positively Skewed dan berbentuk runcing sehingga bersifat Leptokurtis.

```
[29]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot
BoxPlotKeliling= sns.boxplot(data=Keliling)
BoxPlotKeliling.set_title("BoxPlot Keliling")
```

```
[29]: Text(0.5, 1.0, 'BoxPlot Keliling')
```

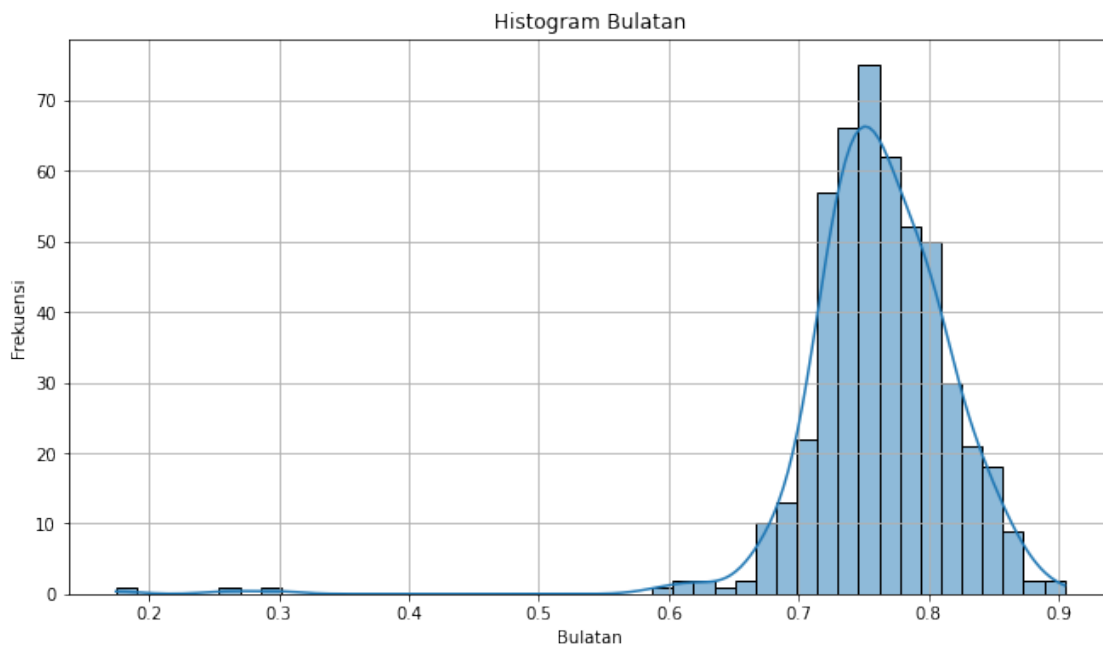


Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom Keliling sekitar 200, nilai Q1 sekitar 260, nilai Q2 atau median sekitar 280, nilai Q3 sekitar 310, dan nilai maksimal sekitar 490. Boxplot ini juga menunjukan bahwa kolom Keliling memiliki 4 pencilan atas dengan nilai sekitar 390, 440, 450, dan 490.

## 5.10 10. Bulatan

```
[30]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistBulatan =sns.histplot(Bulatan, kde=True)
HistBulatan.set_title("Histogram Bulatan")
HistBulatan.set_ylabel("Frekuensi")
```

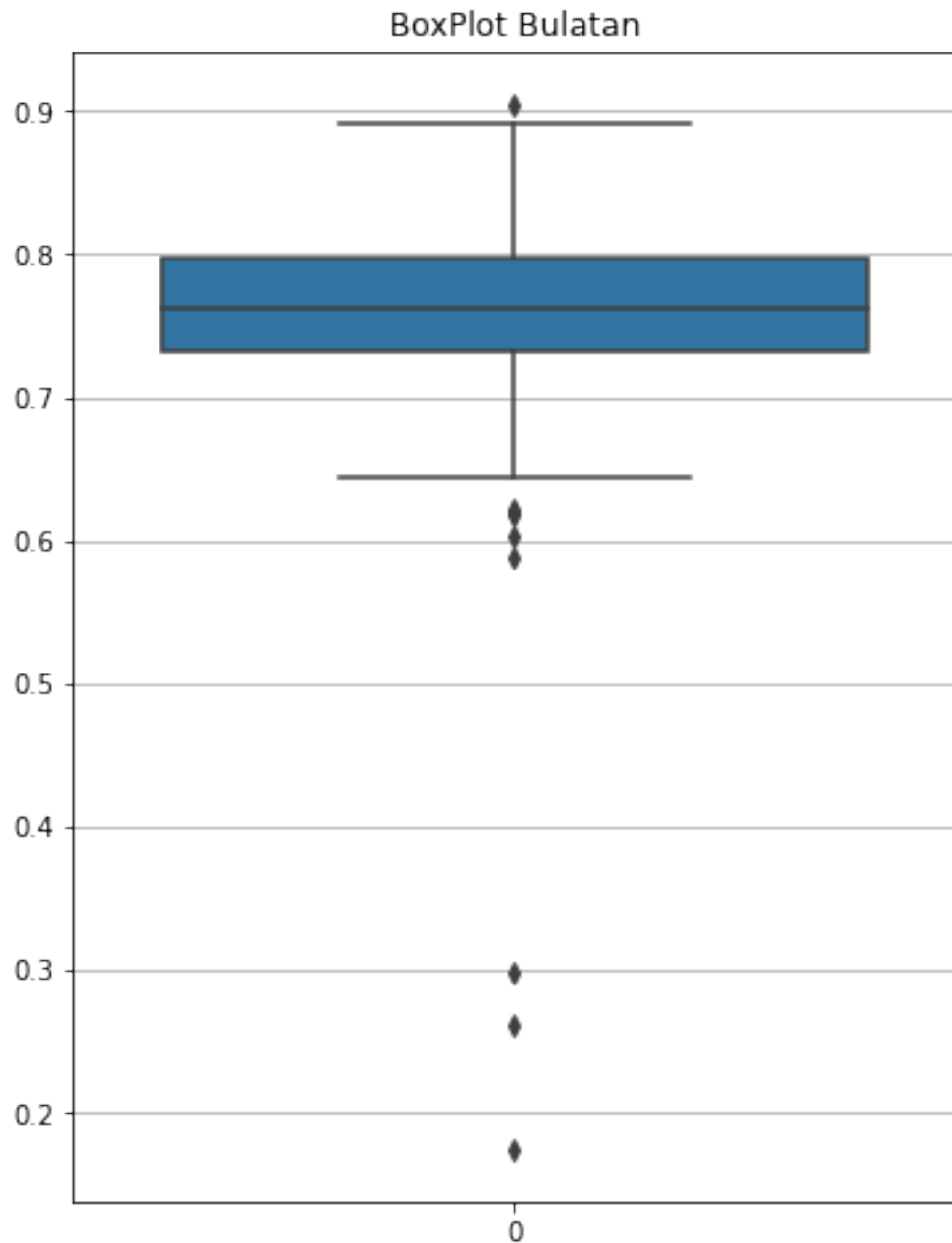
```
[30]: Text(0, 0.5, 'Frekuensi')
```



Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom Bulatan terdistribusi secara condong ke kiri sehingga bersifat Negatively Skewed dan berbentuk runcing sehingga bersifat Leptokurtis.

```
[31]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot
BoxPlotBulatan= sns.boxplot(data=Bulatan)
BoxPlotBulatan.set_title("BoxPlot Bulatan")
```

```
[31]: Text(0.5, 1.0, 'BoxPlot Bulatan')
```

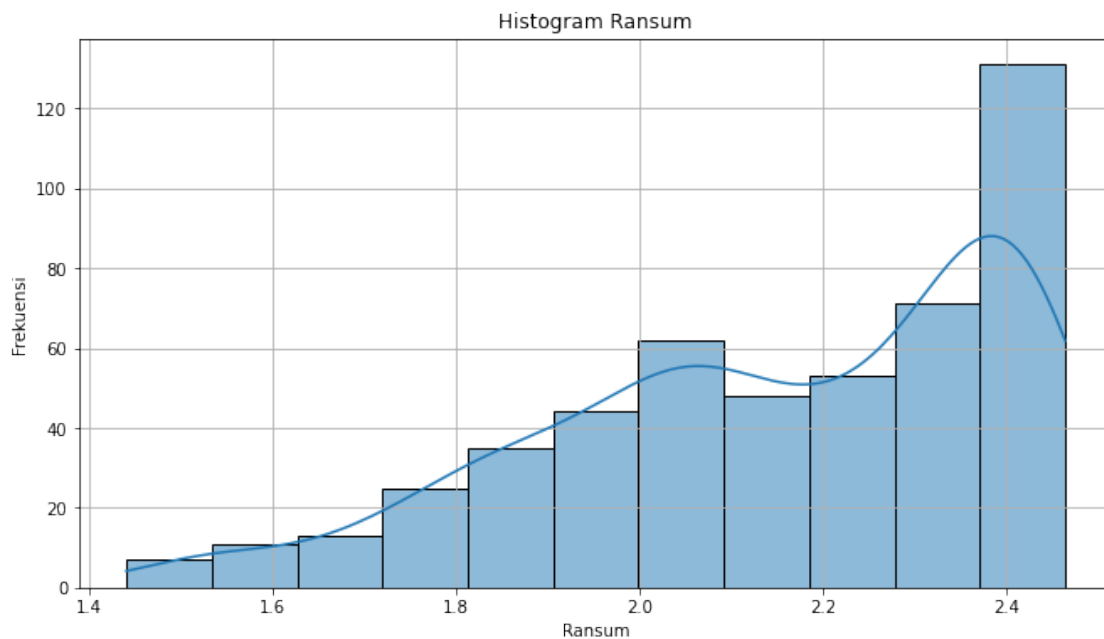


Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom Bulatan sekitar 0.18, nilai Q1 sekitar 0.74, nilai Q2 atau median sekitar 0.76, nilai Q3 sekitar 0.8, dan nilai maksimal sekitar 0.9. Boxplot ini juga menunjukkan bahwa kolom Bulatan memiliki beberapa pencilan yaitu 6 pencilan bawah dengan nilai dibawah 0.65 dan 1 pencilan atas dengan nilai sekitar 0.9.

## 5.11 11. Ransum

```
[32]: # Konfigurasi Histogram
plt.figure(figsize=(11,6))
plt.grid()
# Penggambaran Histogram
HistRansum =sns.histplot(Ransum, kde=True)
HistRansum.set_title("Histogram Ransum")
HistRansum.set_ylabel("Frekuensi")
```

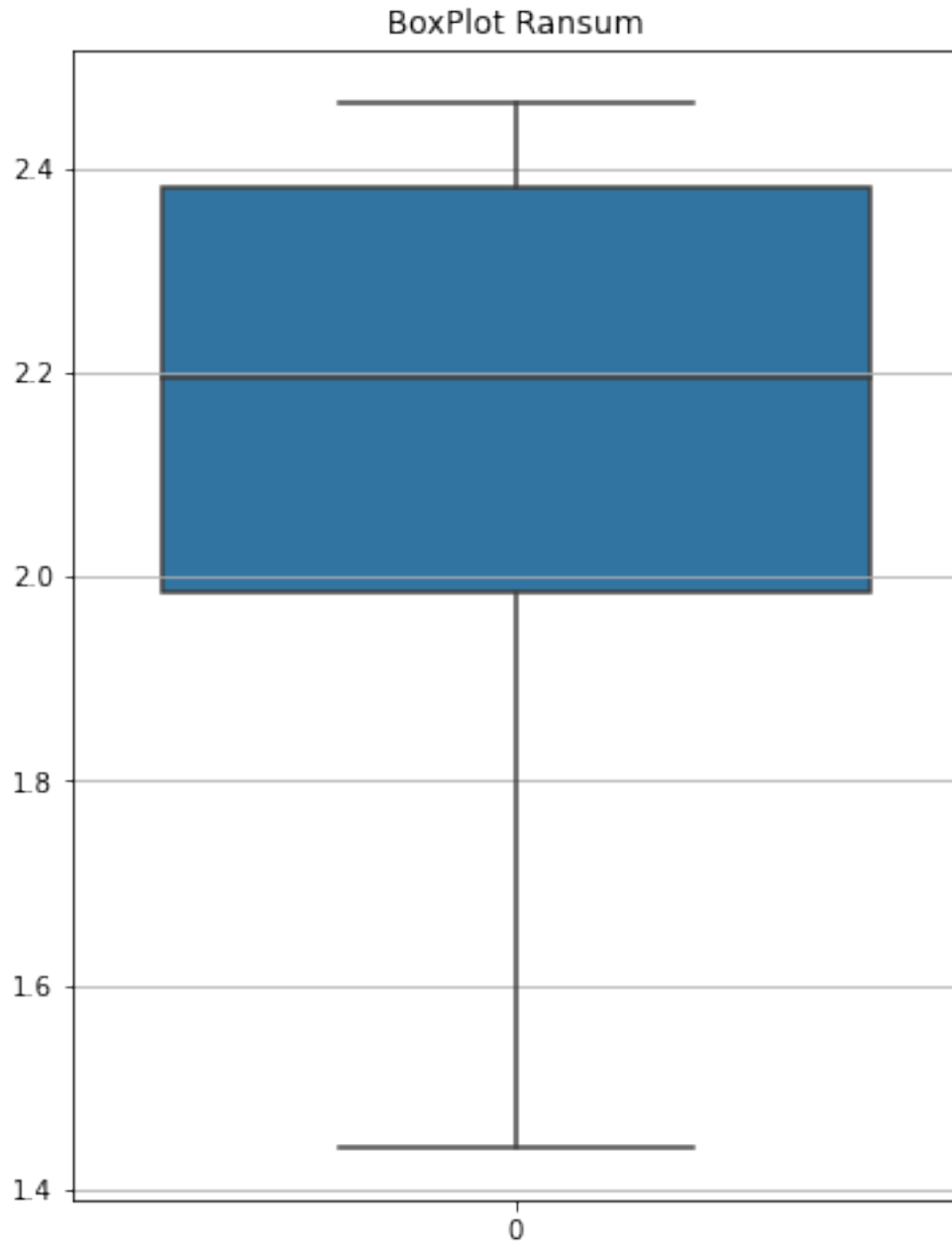
```
[32]: Text(0, 0.5, 'Frekuensi')
```



Berdasarkan histogram diatas, didapatkan sebuah informasi bahwa data pada kolom Ransum terdistribusi secara condong ke kiri sehingga bersifat Negatively Skewed dan berbentuk tidak runcing dan tidak landai sehingga bersifat Mesokurtis.

```
[33]: # Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot
BoxPlotRansum= sns.boxplot(data=Ransum)
BoxPlotRansum.set_title("BoxPlot Ransum")
```

```
[33]: Text(0.5, 1.0, 'BoxPlot Ransum')
```



Berdasarkan Boxplot diatas dapat diketahui bahwa nilai minimum dari kolom Ransum sekitar 1.45, nilai Q1 sekitar 1.98, nilai Q2 atau median sekitar 2.2, nilai Q3 sekitar 2.38, dan nilai maksimal sekitar 2.46. Boxplot ini juga menunjukan bahwa kolom Ransum tidak memiliki pencilan.

### 5.12 12. Kelas

Nilai-nilai pada kolom kelas merepresentasikan jenis dari suatu gandum (bukan numerik)



## 6 Section 3 - Normality Test

Menentukan setiap kolom numerik berdistribusi normal atau tidak. Gunakan normality test yang dikaitkan dengan histogram plot.

### 6.0.1 1. Id

id hanyalah pembeda untuk setiap row, sehingga tidak perlu dilakukan normality test terhadap kolom id sekalipun kolom tersebut numerik

### 6.0.2 2. Daerah

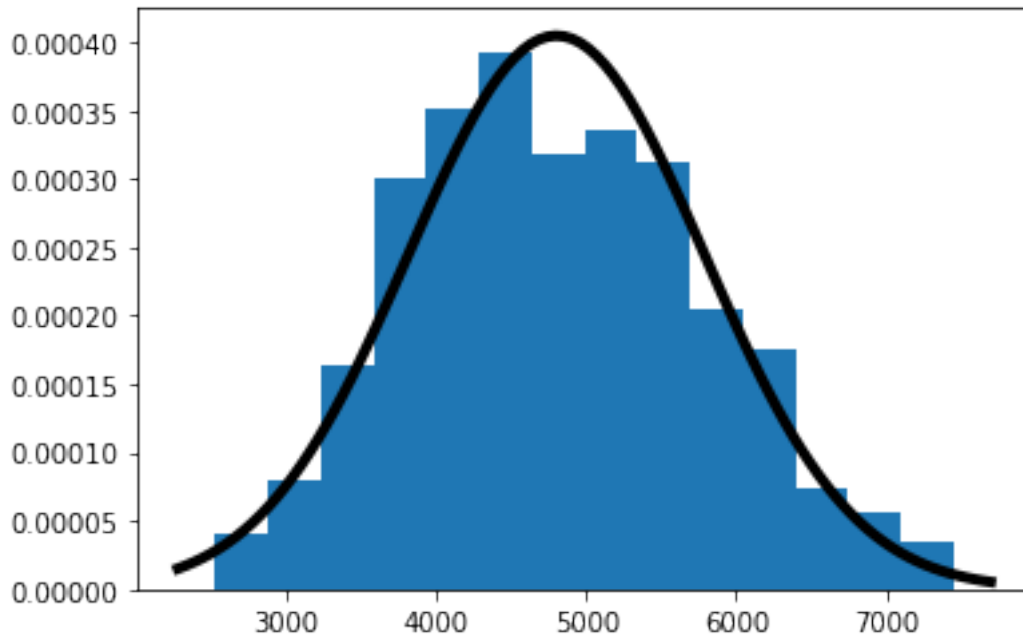
```
[34]: k2, p = st.normaltest(Daerah)
alpha = 0.05
print("p = {:g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

mean, std = st.norm.fit(Daerah)
plt.hist(Daerah, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k', lw=4)
```

```
p = 0.00438627
```

```
H0 ditolak, sehingga data tidak terdistribusi dengan normal
```

```
[34]: [<matplotlib.lines.Line2D at 0x21c22809eb0>]
```



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan terdistribusi dengan normal karena bersifat No Skew. Tapi asumsi ini dapat dibantah karena berdasarkan Normality Test didapatkan kesimpulan data tidak terdistribusi dengan normal. Sehingga kesimpulan data pada kolom Daerah tidak terdistribusi secara normal.

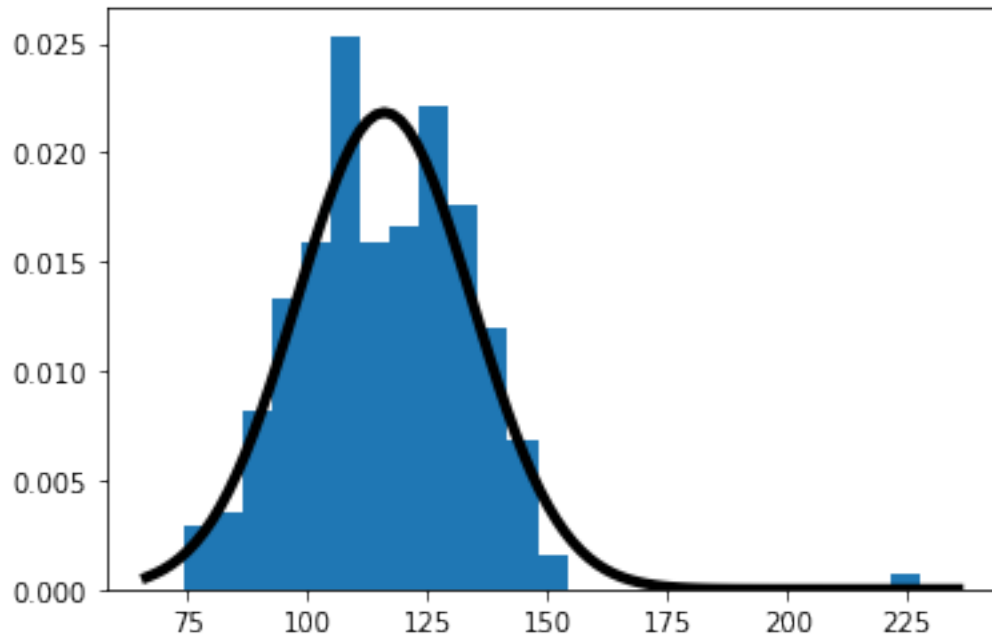
### 6.0.3 3. SumbuUtama

```
[35]: k2, p = st.normaltest(SumbuUtama)
alpha = 0.05
print("p = {:g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

mean, std = st.norm.fit(SumbuUtama)
plt.hist(SumbuUtama, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k', lw=4)
```

```
p = 2.21274e-21
H0 ditolak, sehingga data tidak terdistribusi dengan normal
```

```
[35]: [<matplotlib.lines.Line2D at 0x21c22894910>]
```



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan tidak terdistribusi dengan normal karena histogram condong ke kanan (Positively Skewed). Asumsi ini diperkuat dengan hasil Normality Test yang menghasilkan kesimpulan bahwa data tidak terdistribusi dengan normal. Sehingga kesimpulannya data pada kolom SumbuUtama tidak terdistribusi secara normal.

#### 6.0.4 4. SumbuKecil

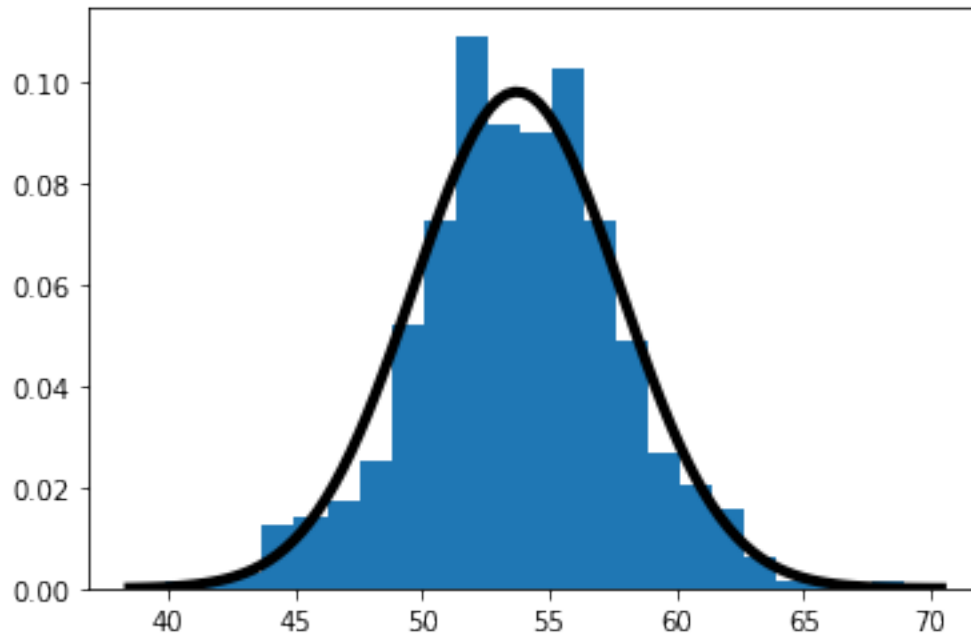
```
[36]: k2, p = st.normaltest(SumbuKecil)
alpha = 0.05
print("p = {:g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

mean,std=st.norm.fit(SumbuKecil)
plt.hist(SumbuKecil, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k',lw=4)
```

p = 0.157363

H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal

[36]: [<matplotlib.lines.Line2D at 0x21c22934580>]



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan terdistribusi dengan normal karena bersifat No Skew. Asumsi ini diperkuat dengan hasil Normality Test yang menghasilkan kesimpulan bahwa data terdistribusi dengan normal. Sehingga kesimpulannya data pada kolom SumbuKecil terdistribusi secara normal.

#### 6.0.5 5. Keunikan

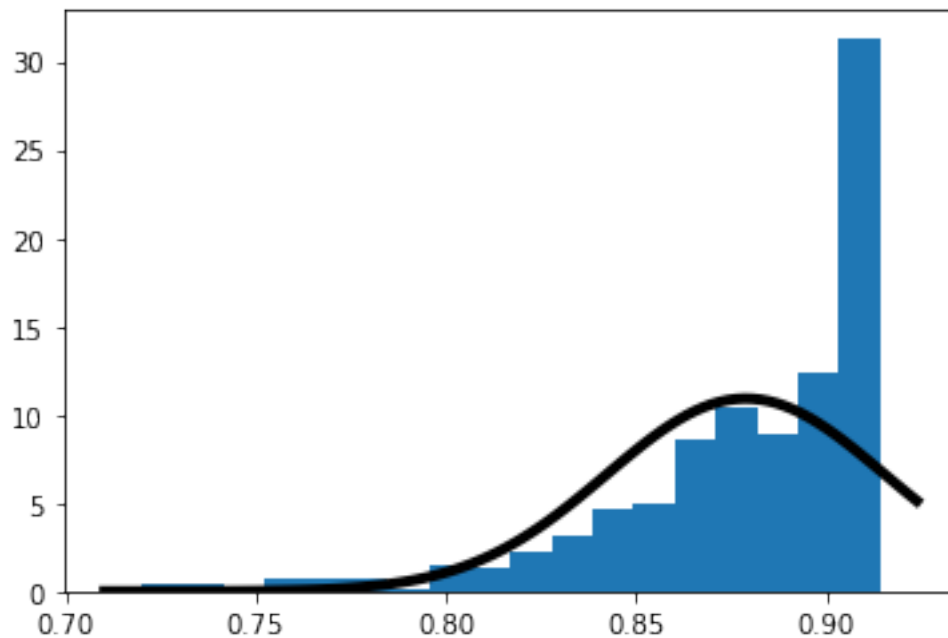
```
[37]: k2, p = st.normaltest(Keunikan)
alpha = 0.05
print("p = {:g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

mean, std = st.norm.fit(Keunikan)
plt.hist(Keunikan, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k', lw=4)
```

p = 3.60297e-35

H0 ditolak, sehingga data tidak terdistribusi dengan normal

[37]: [<matplotlib.lines.Line2D at 0x21c229c5dc0>]



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan tidak terdistribusi dengan normal karena histogram condong ke kiri (Negatively Skewed). Asumsi ini diperkuat dengan hasil Normality Test yang menghasilkan kesimpulan bahwa data tidak terdistribusi dengan normal. Sehingga kesimpulannya data pada kolom Keunikan tidak terdistribusi secara normal.

#### 6.0.6 6. AreaBulatan

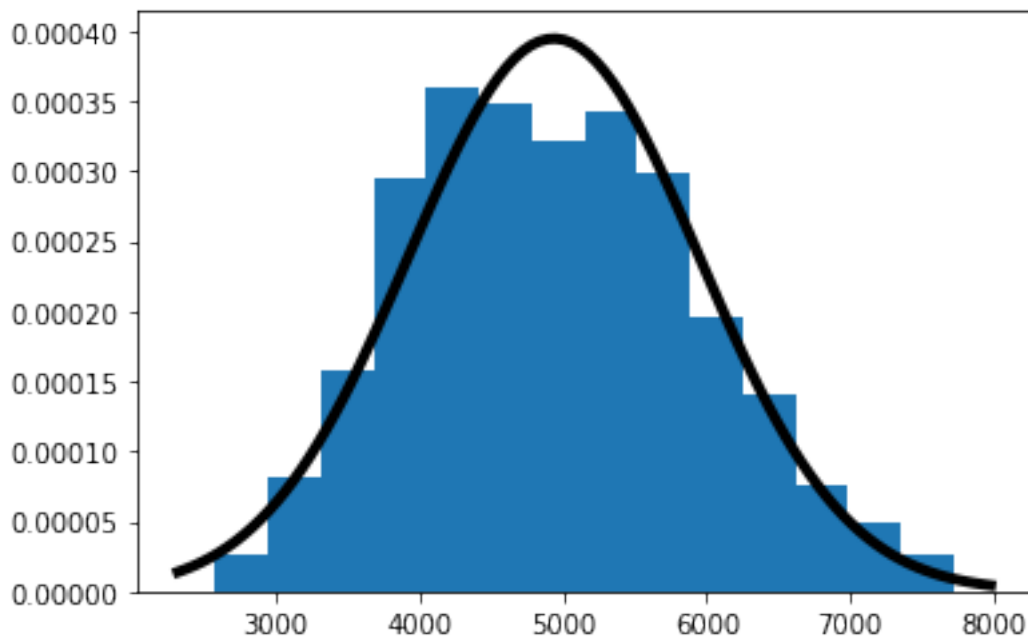
```
[38]: k2, p = st.normaltest(AreaBulatan)
alpha = 0.05
print("p = {:g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

mean, std = st.norm.fit(AreaBulatan)
plt.hist(AreaBulatan, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k', lw=4)
```

p = 0.00465706

H0 ditolak, sehingga data tidak terdistribusi dengan normal

[38]: [<matplotlib.lines.Line2D at 0x21c22a55670>]



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan terdistribusi dengan normal karena bersifat No Skew. Tapi asumsi ini dapat dibantah karena berdasarkan Normality Test didapatkan kesimpulan data tidak terdistribusi dengan normal. Sehingga kesimpulannya data pada kolom AreaBulatan tidak terdistribusi secara normal.

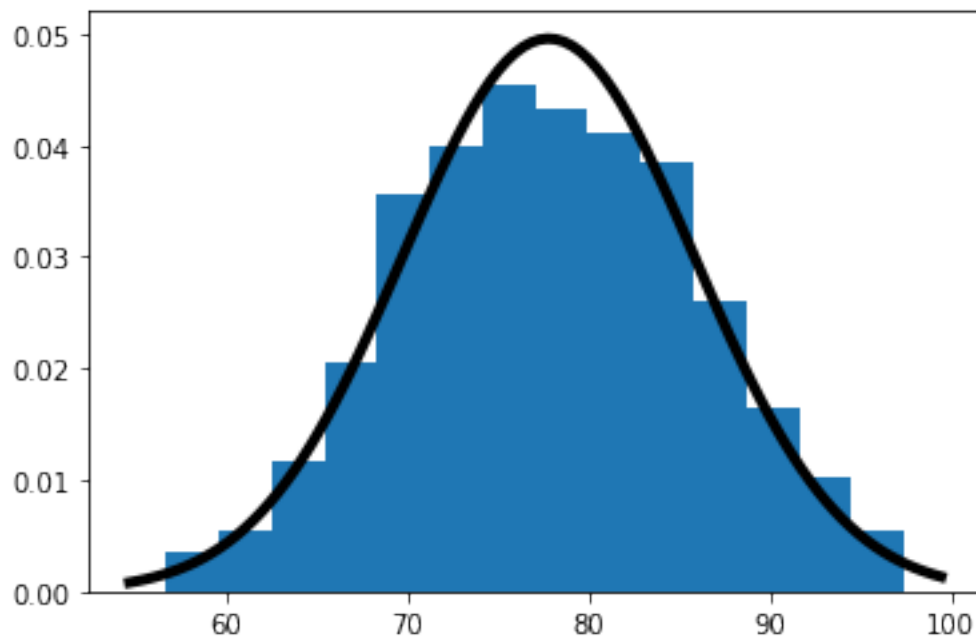
#### 6.0.7 7. Diameter

```
[39]: k2, p = st.normaltest(Diameter)
alpha = 0.05
print("p = {:.g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

mean,std=st.norm.fit(Diameter)
plt.hist(Diameter, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k',lw=4)
```

```
p = 0.0241572
H0 ditolak, sehingga data tidak terdistribusi dengan normal
```

[39]: [<matplotlib.lines.Line2D at 0x21c22ad97f0>]



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan terdistribusi dengan normal karena bersifat No Skew. Tapi asumsi ini dapat dibantah karena berdasarkan Normality Test didapatkan kesimpulan data tidak terdistribusi dengan normal. Sehingga kesimpulannya data pada kolom Diameter tidak terdistribusi secara normal.

## 6.0.8 8. KadarAir

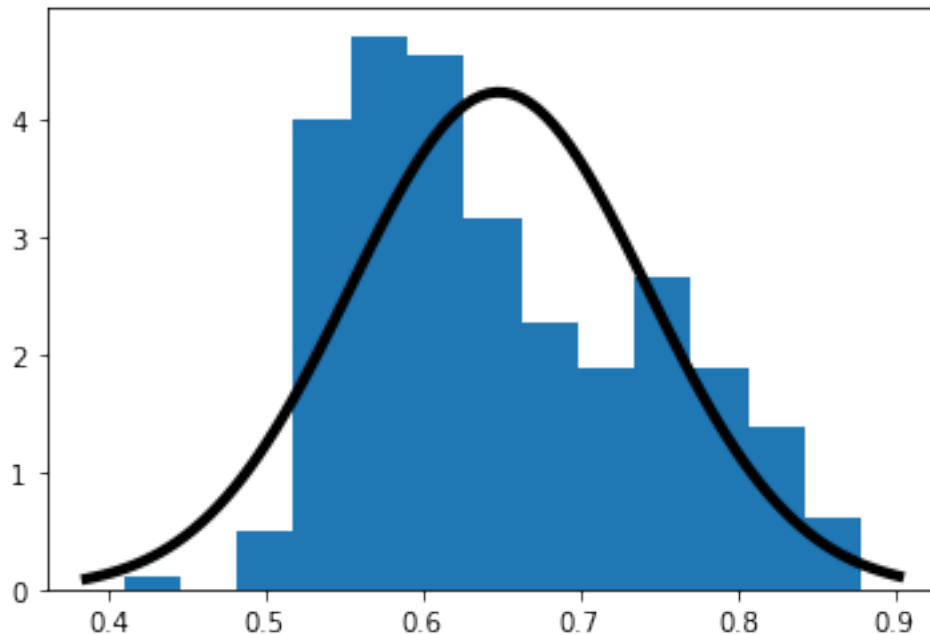
```
[40]: k2, p = st.normaltest(KadarAir)
alpha = 0.05
print("p = {:g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

mean,std=st.norm.fit(KadarAir)
plt.hist(KadarAir, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k',lw=4)
```

p = 2.58694e-11

H0 ditolak, sehingga data tidak terdistribusi dengan normal

[40]: [<matplotlib.lines.Line2D at 0x21c23b2f130>]



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan terdistribusi dengan normal karena bersifat No Skew walaupun mendekati condong ke kanan. Tapi asumsi ini dapat dibantah karena berdasarkan Normality Test didapatkan kesimpulan data tidak terdistribusi dengan normal. Sehingga kesimpulannya data pada kolom KadarAir tidak terdistribusi secara normal.

### 6.0.9 9. Keliling

```
[41]: k2, p = st.normaltest(Keliling)
alpha = 0.05
print("p = {:g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

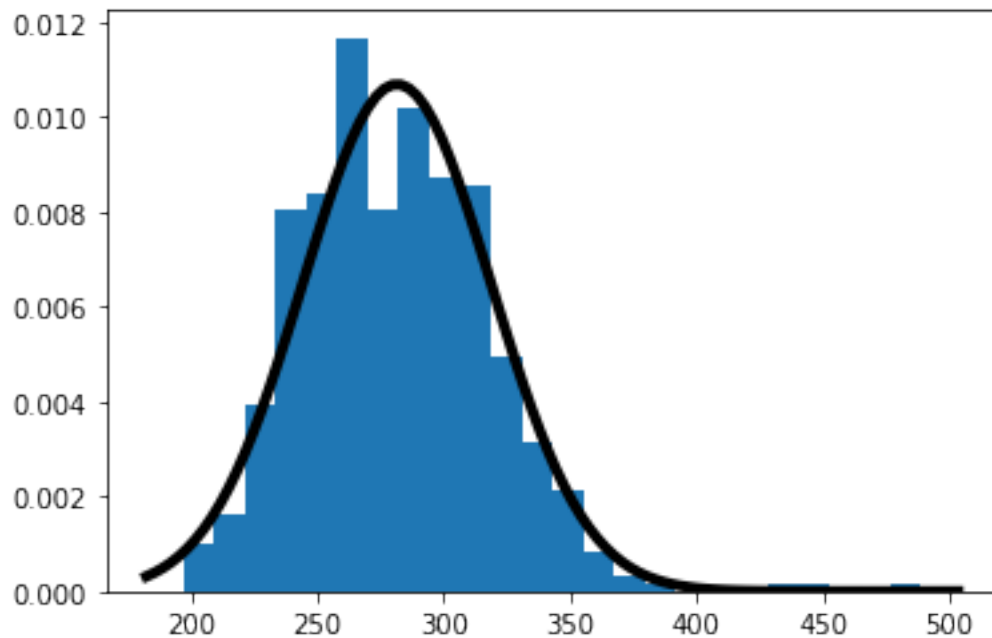
mean,std=st.norm.fit(Keliling)
plt.hist(Keliling, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k',lw=4)
```

p = 2.23901e-15

H0 ditolak, sehingga data tidak terdistribusi dengan normal



[41]: [<matplotlib.lines.Line2D at 0x21c23ba38e0>]



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan tidak terdistribusi dengan normal karena histogram condong ke kanan (Positively Skewed). Asumsi ini diperkuat dengan hasil Normality Test yang menghasilkan kesimpulan bahwa data tidak terdistribusi dengan normal. Sehingga kesimpulannya data pada kolom Keliling tidak terdistribusi secara normal.

#### 6.0.10 10. Bulatan

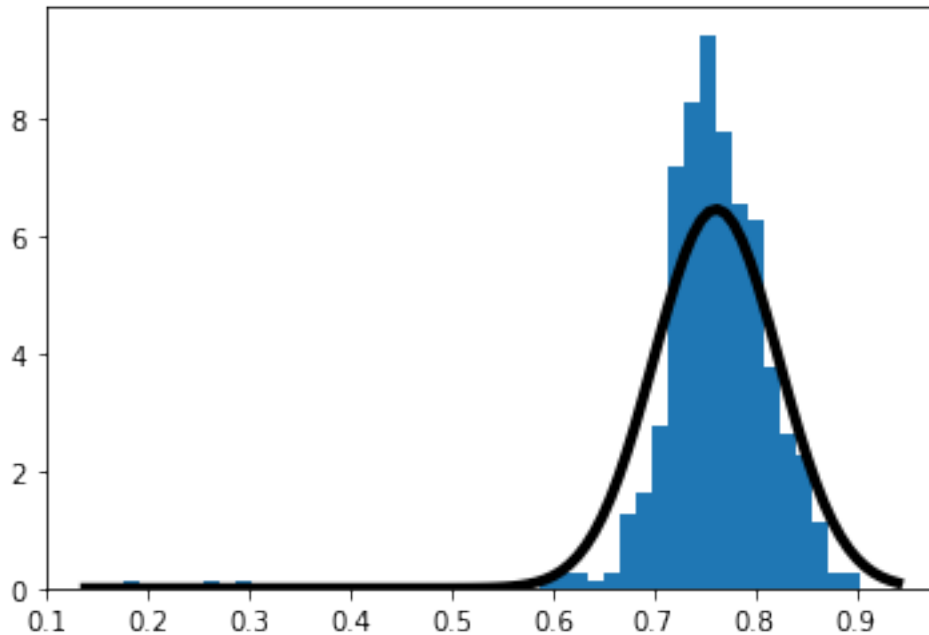
```
[42]: k2, p = st.normaltest(Bulatan)
alpha = 0.05
print("p = {:g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

mean,std=st.norm.fit(Bulatan)
plt.hist(Bulatan, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k',lw=4)
```

p = 9.88528e-97

H0 ditolak, sehingga data tidak terdistribusi dengan normal

[42]: [<matplotlib.lines.Line2D at 0x21c23c3d9d0>]



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan tidak terdistribusi dengan normal karena histogram condong ke kiri (Negatively Skewed). Asumsi ini diperkuat dengan hasil Normality Test yang menghasilkan kesimpulan bahwa data tidak terdistribusi dengan normal. Sehingga kesimpulannya data pada kolom Bulatan tidak terdistribusi secara normal.

#### 6.0.11 11. Ransum

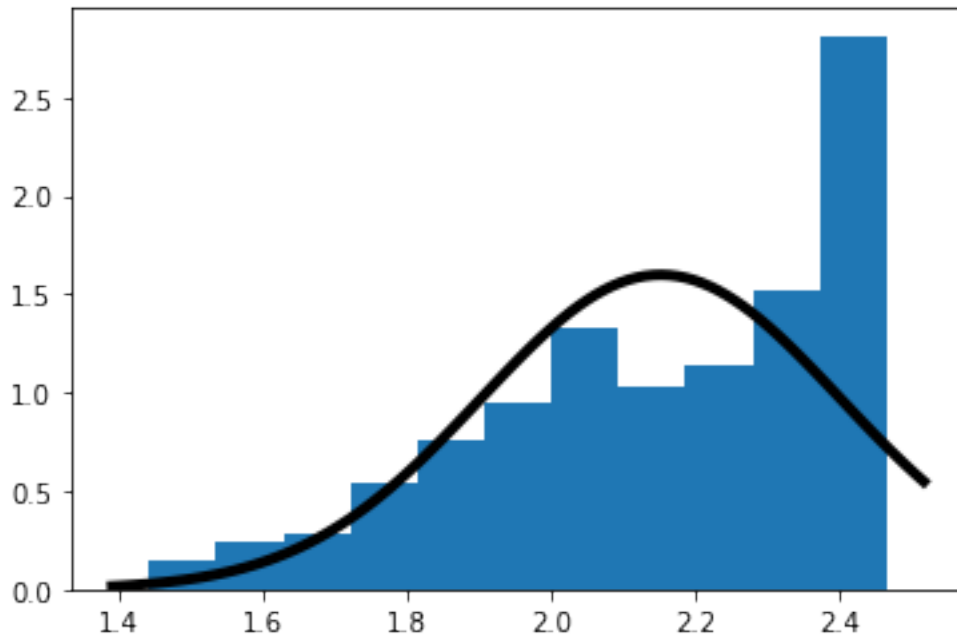
```
[43]: k2, p = st.normaltest(Ransum)
alpha = 0.05
print("p = {:g}".format(p))
if p < alpha: # null hypothesis: x comes from a normal distribution
    print("H0 ditolak, sehingga data tidak terdistribusi dengan normal")
else:
    print("H0 tidak bisa ditolak, sehingga data terdistribusi dengan normal")

mean, std = st.norm.fit(Ransum)
plt.hist(Ransum, bins='auto', density=True)
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
y = st.norm.pdf(x, mean, std)
plt.plot(x, y, 'k', lw=4)
```

p = 8.96301e-09

H0 ditolak, sehingga data tidak terdistribusi dengan normal

[43]: [<matplotlib.lines.Line2D at 0x21c23d15e50>]



Berdasarkan histogram plotnya terlihat bahwa histogram dapat diasumsikan tidak terdistribusi dengan normal karena histogram condong ke kiri (Negatively Skewed). Asumsi ini diperkuat dengan hasil Normality Test yang menghasilkan kesimpulan bahwa data tidak terdistribusi dengan normal. Sehingga kesimpulannya data pada kolom Ransum tidak terdistribusi secara normal.

#### 6.0.12 12. Kelas

Nilai-nilai pada kolom kelas merepresentasikan jenis dari suatu gandum (bukan numerik)

## 7 Section 4 - Tes Hipotesis 1 Sampel

Melakukan test hipotesis 1 sampel, dengan menuliskan 6 langkah testing dan menampilkan juga boxplotnya untuk kolom/bagian yang bersesuaian. **Enam Langkah Testing:** 1. Tentukan Hipotesis nol ( $H_0: \theta = \theta_0$ ), dimana  $\theta$  bisa berupa  $\mu$ ,  $\sigma^2$ ,  $p$ , atau data lain berdistribusi tertentu (normal, binomial, dsc.). 2. Pilih hipotesis alternatif  $H_1$  salah dari  $\theta > \theta_0$ ,  $\theta < \theta_0$ , atau  $\theta \neq \theta_0$ . 3. Tentukan tingkat signifikan  $\alpha$ . 4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis. 5. Hitung nilai uji statistik dari data sample. Hitung p-value sesuai dengan uji statistik yang digunakan. 6. Ambil keputusan dengan TOLAK  $H_0$  jika nilai uji terletak di daerah kritis atau dengan tes signifikan, TOLAK  $H_0$  jika p-value lebih kecil dibanding tingkat signifikansi  $\alpha$  yang diinginkan

```
[44]: def Z_testStatistic(x̄, μ0, σ, root_n):  
      return (float)(x̄ - μ0) / (σ / root_n)  
      def Z_testStatistic_bigN(p̂, p0, q0, n):
```

```

    return (float)(p^-p0)/np.sqrt(p0*q0/n)
def T_testStatistic(x-,μ0,s,root_n):
    return (float)(x--μ0)/(s/root_n)

```

## 7.1 A. Nilai rata-rata Daerah di atas 4700?

Asumsi sampel yang dicek sejumlah 500 dimana 500 data tersebut juga merupakan populasi (sampel yang dicek sekaligus populasi). Karena itu digunakan uji statistik distribusi Z untuk satu mean dengan standar deviasi populasi diketahui.

```

[45]: print("Nilai rata-rata Daerah di atas 4700?")

# Langkah 1
H0 = "μ=4700"
print("1. H0 : {}".format(H0))

# Langkah 2
H1 = "μ>4700"
print("2. H1 : {}".format(H1))

# Langkah 3
α = 5e-2
print("3. α = {}".format(α))

# Langkah 4
zα = round(st.norm.ppf(1-α),3)
print("4. Uji Statistik : z=(x--μ0)/(σ/root_n), σ diketahui")
print("    Daerah Kritis : z>zα : z >{}".format(zα))

# Langkah 5
x- = Daerah.mean()
μ0= 4700
σ = Daerah.std()
root_n = np.sqrt(len(Daerah))
z = round(Z_testStatistic(x-,μ0,σ,root_n),3)
p_value = 1-st.norm.cdf(z)
print("5. Komputasi")
print("    x- : {} \n    root_n: {} \n    σ: {} \n    μ0 : {}".
      →format(x-,root_n,σ,μ0))
print("    p_value : {} \n    z: {}".format(str(p_value),str(z)))

# Langkah 6
print("6. Test Daerah Kritis")
if (z > zα):
    print("    Tolak H0 karena nilai uji = {}>{} (z>zα)".format(str(z),str(zα)))
    print("    Rata-Rata daerah di atas 4700")
else :

```

```

    print("    Terima H0 karena nilai uji = {}<={} (z<=z $\alpha$ )".
    →format(str(z),str(z $\alpha$ )))
    print("    Rata-Rata daerah sama dengan 4700")

# Menggambar Boxplot Daerah

# Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()

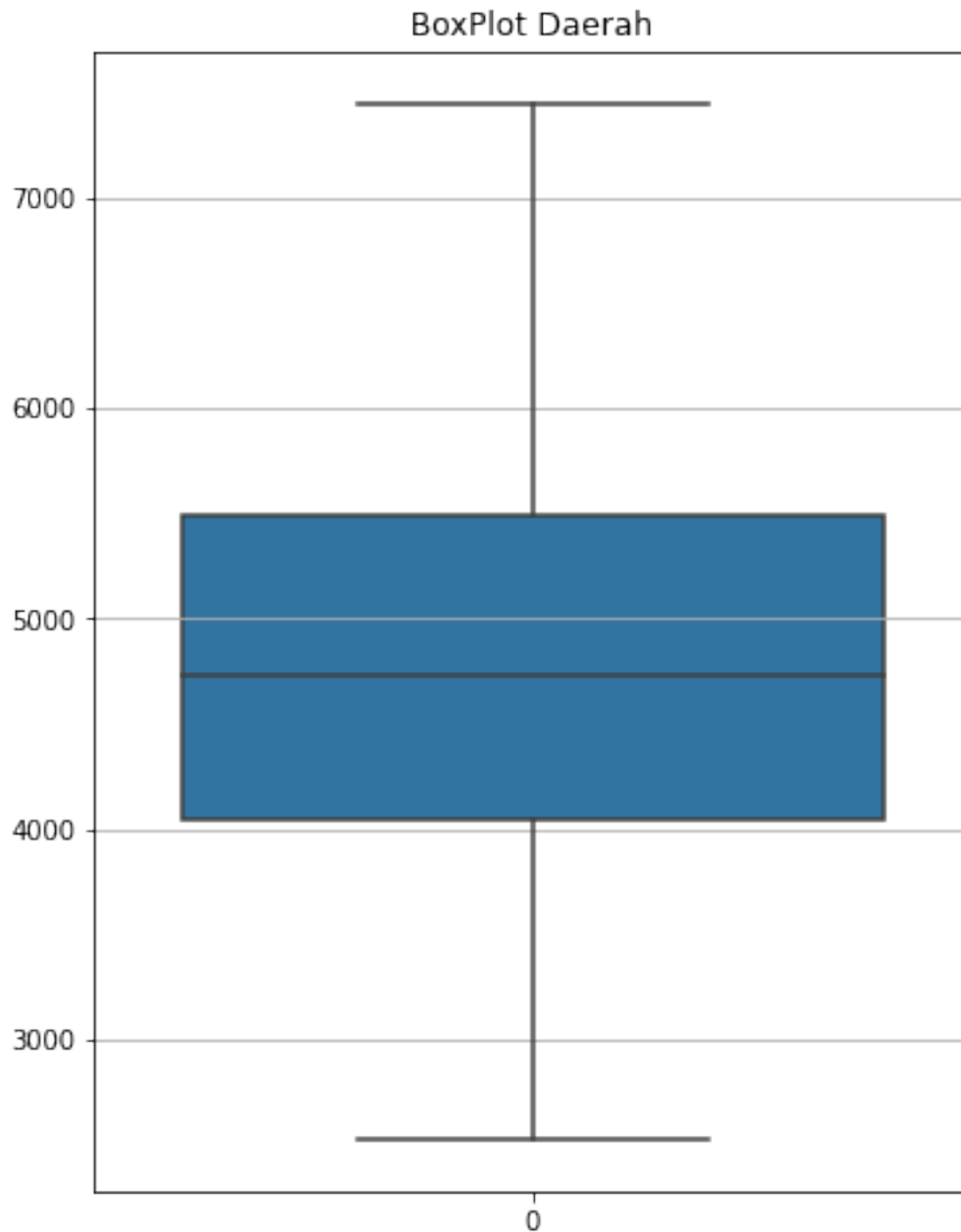
# Penggambaran Boxplot
BoxPlotDaerah= sns.boxplot(data=Daerah)
BoxPlotDaerah.set_title("BoxPlot Daerah")

```

Nilai rata-rata Daerah di atas 4700?

1. H0 :  $\mu=4700$
2. H1 :  $\mu>4700$
3.  $\alpha = 0.05$
4. Uji Statistik :  $z=(\bar{x}-\mu_0)/(\sigma/\text{root\_n})$ ,  $\sigma$  diketahui  
Daerah Kritis :  $z>z\alpha$  :  $z >1.645$
5. Komputasi  
 $\bar{x}$  : 4801.246  
 root\_n: 22.360679774997898  
 $\sigma$ : 986.3954914816017  
 $\mu_0$  : 4700  
 p\_value : 0.010866562148332704  
 z: 2.295
6. Test Daerah Kritis  
 Tolak H0 karena nilai uji = 2.295>1.645 ( $z>z\alpha$ )  
 Rata-Rata daerah di atas 4700

[45]: Text(0.5, 1.0, 'BoxPlot Daerah')



## 7.2 B. Nilai Rata-rata Sumbu Utama tidak sama dengan 116?

Asumsi sampel yang dicek sejumlah 500 dimana 500 data tersebut juga merupakan populasi (sampel yang dicek sekaligus populasi). Karena itu digunakan uji statistik distribusi Z untuk satu mean dengan standar deviasi populasi diketahui.

```
[46]: print("Nilai Rata-rata Sumbu Utama tidak sama dengan 116?")
```

```
# Langkah 1
```

```

H0 = " $\mu=116$ "
print("1. H0 : {}".format(H0))

# Langkah 2
H1 = " $\mu \neq 116$ "
print("2. H1 : {}".format(H1))

# Langkah 3
 $\alpha$  = 5e-2
print("3.  $\alpha$  = {}".format( $\alpha$ ))

# Langkah 4
z $\alpha$ _div2 = round(st.norm.ppf(1-( $\alpha$ /2)),3)
print("4. Uji Statistik :  $z=(\bar{x}-\mu_0)/(\sigma/\text{root\_n})$ ,  $\sigma$  diketahui")
print("    Daerah Kritis :  $z > z\alpha/2$  atau  $z < -z\alpha/2$  :  $z > {}$  atau  $z < {}$ ".
      →format(z $\alpha$ _div2,-1*z $\alpha$ _div2))

# Langkah 5
x $^-$  = SumbuUtama.mean()
 $\mu_0$ = 116
 $\sigma$  = SumbuUtama.std()
n = len(SumbuUtama)
root_n = np.sqrt(n)
z = round(Z_testStatistic(x $^-$ , $\mu_0$ , $\sigma$ ,root_n),3)
p_value = 1-abs(st.norm.cdf(z)-st.norm.cdf(-1*z))
print("5. Komputasi")
print("    x $^-$  : {} \n    n: {} \n    root_n: {} \n     $\sigma$ : {} \n     $\mu_0$  : {}".
      →format(x $^-$ ,n,root_n, $\sigma$ , $\mu_0$ ))
print("    p_value : {} \n    z: {}".format(str(p_value),str(z)))

# Langkah 6
print("6. Test Daerah Kritis")
if (z > z $\alpha$ _div2 or z < -1*z $\alpha$ _div2):
    if (z > z $\alpha$ _div2):
        print("    Tolak H0 karena nilai uji = {}>{} ( $z > z\alpha/2$ )".format(z,z $\alpha$ _div2))
    else:
        print("    Tolak H0 karena nilai uji = {}<{} ( $z < -z\alpha/2$ )".
              →format(z,-1*z $\alpha$ _div2))
        print("    Rata-rata Sumbu Utama tidak sama dengan 116")
else :
    print("    Terima H0 karena nilai uji = {}<{}<{} ( $-z\alpha/2 < z < z\alpha/2$ )".
          →format(-1*z $\alpha$ _div2,z,z $\alpha$ _div2))
    print("    Rata-rata Sumbu Utama sama dengan 116")

# Menggambar Boxplot SumbuUtama

# Konfigurasi Boxplot

```

```
plt.figure(figsize=(6,8))
plt.grid()

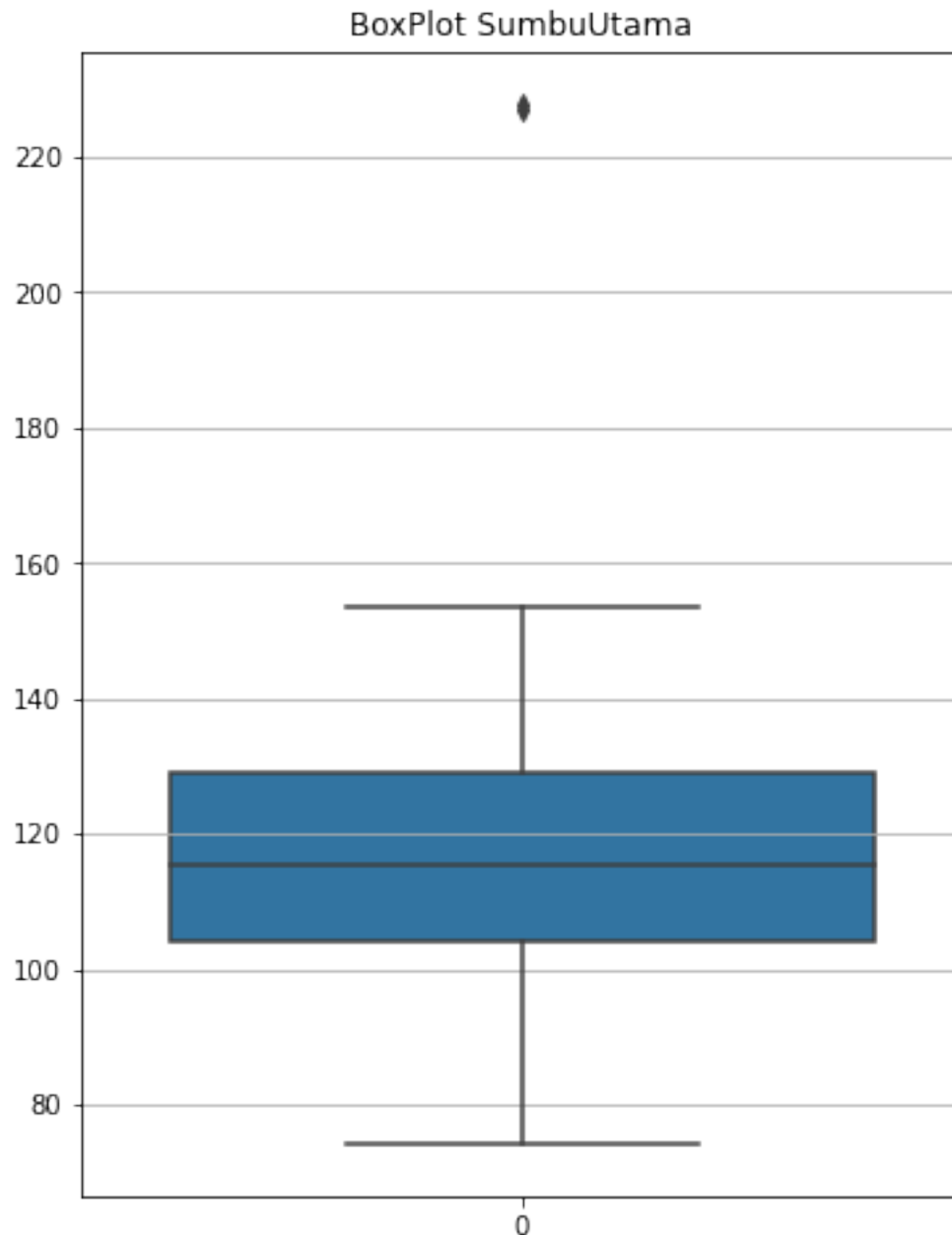
# Penggambaran Boxplot
BoxPlotDaerah= sns.boxplot(data=SumbuUtama)
BoxPlotDaerah.set_title("BoxPlot SumbuUtama")
```

Nilai Rata-rata Sumbu Utama tidak sama dengan 116?

1.  $H_0 : \mu=116$
2.  $H_1 : \mu \neq 116$
3.  $\alpha = 0.05$
4. Uji Statistik :  $z=(\bar{x}-\mu_0)/(\sigma/\text{root\_n})$ ,  $\sigma$  diketahui  
Daerah Kritis :  $z>z\alpha/2$  atau  $z <-z\alpha/2$  :  $z>1.96$  atau  $z<-1.96$
5. Komputasi  
 $\bar{x}$  : 116.04517136778  
 n: 500  
 root\_n: 22.360679774997898  
 $\sigma$ : 18.28262595755935  
 $\mu_0$  : 116  
 p\_value : 0.9561384637943531  
 z: 0.055
6. Test Daerah Kritis  
 Terima  $H_0$  karena nilai uji =  $-1.96 < 0.055 < 1.96$  ( $-z\alpha/2 < z < z\alpha/2$ )  
 Rata-rata Sumbu Utama sama dengan 116

[46]: Text(0.5, 1.0, 'BoxPlot SumbuUtama')





### 7.3 C. Nilai Rata-rata 20 baris pertama kolom Sumbu Kecil bukan 50?

Asumsi sampel yang dicek adalah 20 baris pertama dari populasi yang sejumlah 500 baris. Karena itu digunakan uji statistik distribusi t untuk satu mean dengan standar deviasi populasi yang diketahui.

```
[47]: print("Nilai Rata-rata 20 baris pertama kolom Sumbu Kecil bukan 50?")
      # Langkah 1
      H0 = " $\mu=50$ "
```

```

print("1. H0 : {}".format(H0))

# Langkah 2
H1 = " $\mu \neq 50$ "
print("2. H1 : {}".format(H1))

# Langkah 3
 $\alpha$  = 5e-2
print("3.  $\alpha$  = {}".format( $\alpha$ ))

# Langkah 4
z $\alpha$ _div2 = round(st.norm.ppf(1-( $\alpha$ /2)),3)
print("4. Uji Statistik :  $z = (x^- - \mu_0) / (\sigma / \text{root\_n})$ ,  $\sigma$  diketahui")
print("   Daerah Kritis :  $z > z\alpha/2$  atau  $z < -z\alpha/2$  :  $z > {}$  atau  $z < {}$ ".
       $\rightarrow$ format(z $\alpha$ _div2,z $\alpha$ _div2))

# Langkah 5
newSumbuKecil = SumbuKecil[:20]
x^- = newSumbuKecil.mean()
 $\mu_0$  = 50
 $\sigma$  = SumbuKecil.std()
n = len(newSumbuKecil)
root_n = np.sqrt(n)
z = round(Z_testStatistic(x^-, $\mu_0$ , $\sigma$ ,root_n),3)
p_value = 1-abs(st.norm.cdf(z)-st.norm.cdf(-1*z))
print("5. Komputasi")
print("   x^- : {} \n   root_n: {} \n    $\sigma$ : {} \n    $\mu_0$  : {}".
       $\rightarrow$ format(x^-,root_n, $\sigma$ , $\mu_0$ ))
print("   p_value : {} \n   z: {}".format(str(p_value),str(z)))

# Langkah 6
print("6. Test Daerah Kritis")
if (z > z $\alpha$ _div2 or z < -1*z $\alpha$ _div2):
    if (z > z $\alpha$ _div2):
        print("   Tolak H0 karena nilai uji = {}>{} ( $z > z\alpha/2$ )".format(z,z $\alpha$ _div2))
    else:
        print("   Tolak H0 karena nilai uji = {}<{} ( $z < -z\alpha/2$ )".
               $\rightarrow$ format(z,-1*z $\alpha$ _div2))
        print("   Nilai Rata-rata 20 baris pertama kolom Sumbu Kecil tidak sama_
               $\rightarrow$ dengan 50")
else :
    print("   Terima H0 karena nilai uji = {}<{}<{} ( $-z\alpha/2 < z < z\alpha/2$ )".
           $\rightarrow$ format(-1*z $\alpha$ _div2,z,z $\alpha$ _div2))
    print("   Nilai Rata-rata 20 baris pertama kolom Sumbu Kecil sama dengan 50")

# Konfigurasi Boxplot
plt.figure(figsize=(6,8))

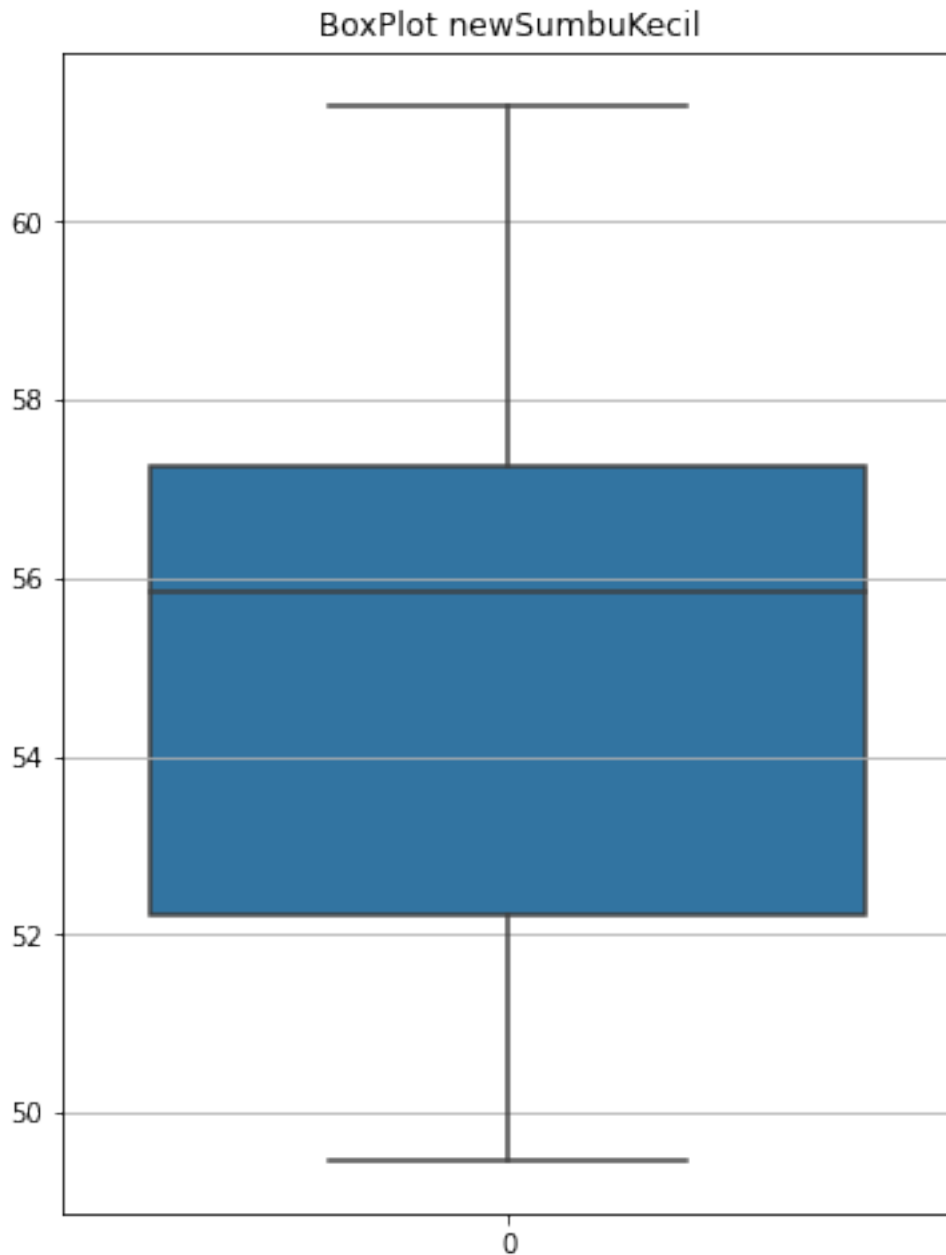
```

```
plt.grid()
# Penggambaran Boxplot
BoxPlotDaerah= sns.boxplot(data=newSumbuKecil)
BoxPlotDaerah.set_title("BoxPlot newSumbuKecil")
```

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

1.  $H_0 : \mu=50$
2.  $H_1 : \mu \neq 50$
3.  $\alpha = 0.05$
4. Uji Statistik :  $z=(\bar{x}-\mu_0)/(\sigma/\text{root\_n})$ ,  $\sigma$  diketahui  
Daerah Kritis :  $z>z\alpha/2$  atau  $z <-z\alpha/2 : z>1.96$  atau  $z<1.96$
5. Komputasi  
 $\bar{x}$  : 54.8872762605  
 root\_n: 4.47213595499958  
 $\sigma$ : 4.0710747524750355  
 $\mu_0$  : 50  
 p\_value : 7.917441946503345e-08  
 z: 5.369
6. Test Daerah Kritis  
 Tolak  $H_0$  karena nilai uji = 5.369>1.96 ( $z>z\alpha/2$ )  
 Nilai Rata-rata 20 baris pertama kolom Sumbu Kecil tidak sama dengan 50

[47]: Text(0.5, 1.0, 'BoxPlot newSumbuKecil')



#### 7.4 D. Proporsi nilai Diameter yang lebih dari 85, adalah tidak sama dengan 15%

Karena sDigunakan uji statistik berupa uji proporsi dengan n besar (binomial didekati normal).

```
[48]: print("Proporsi nilai Diameter yang lebih dari 85, adalah tidak sama dengan 15%")
      # Langkah 1
      H0 = "p=15"
      print("1. H0 : {}".format(H0))
```

```

# Langkah 2
H1 = "p≠15"
print("2. H1 : {}".format(H1))

# Langkah 3
α = 5e-2
print("3. α = {}".format(α))

# Langkah 4

zα_div2 = round(st.norm.ppf(1-(α/2)),3)
print("4. Uji Statistik : z=(p̂-p0)/sqrt(p0*q0/n), σ diketahui")
print("    Daerah Kritis : z>zα/2 atau z <-zα/2 : z>{} atau z<{}".format(zα_div2,zα_div2))

# Langkah 5
newDiameter = [dia for dia in Diameter if dia > 85]
p̂ = len(newDiameter)/len(Diameter)
p0 = 0.15
q0 = 1-p0
n = len(Diameter)
z = round(Z_testStatistic_bigN(p̂,p0,q0,n),3)
p_value = 1-abs(st.norm.cdf(z)-st.norm.cdf(-1*z))
print("5. Komputasi")
print("    p̂ : {} \n    p0: {} \n    q0: {} \n    n : {}".format(p̂,p0,q0,n))
print("    p_value : {} \n    z: {}".format(str(p_value),str(z)))

# Langkah 6
print("6. Test Daerah Kritis")
if (z > zα_div2 or z < -1*zα_div2):
    if (z > zα_div2):
        print("    Tolak H0 karena nilai uji = {}>{} (z>zα/2)".format(z,zα_div2))
    else:
        print("    Tolak H0 karena nilai uji = {}<{} (z<-zα/2)".format(z,-1*zα_div2))
    print("    Proporsi nilai Diameter yang lebih dari 85 tidak sama dengan 15%")
else :
    print("    Terima H0 karena nilai uji = {}<{}<{} (-zα/2<z<zα/2)".format(-1*zα_div2,z,zα_div2))
    print("    Proporsi nilai Diameter yang lebih dari 85 sama dengan 15%")

# Menggambar Boxplot Diameter

# Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()

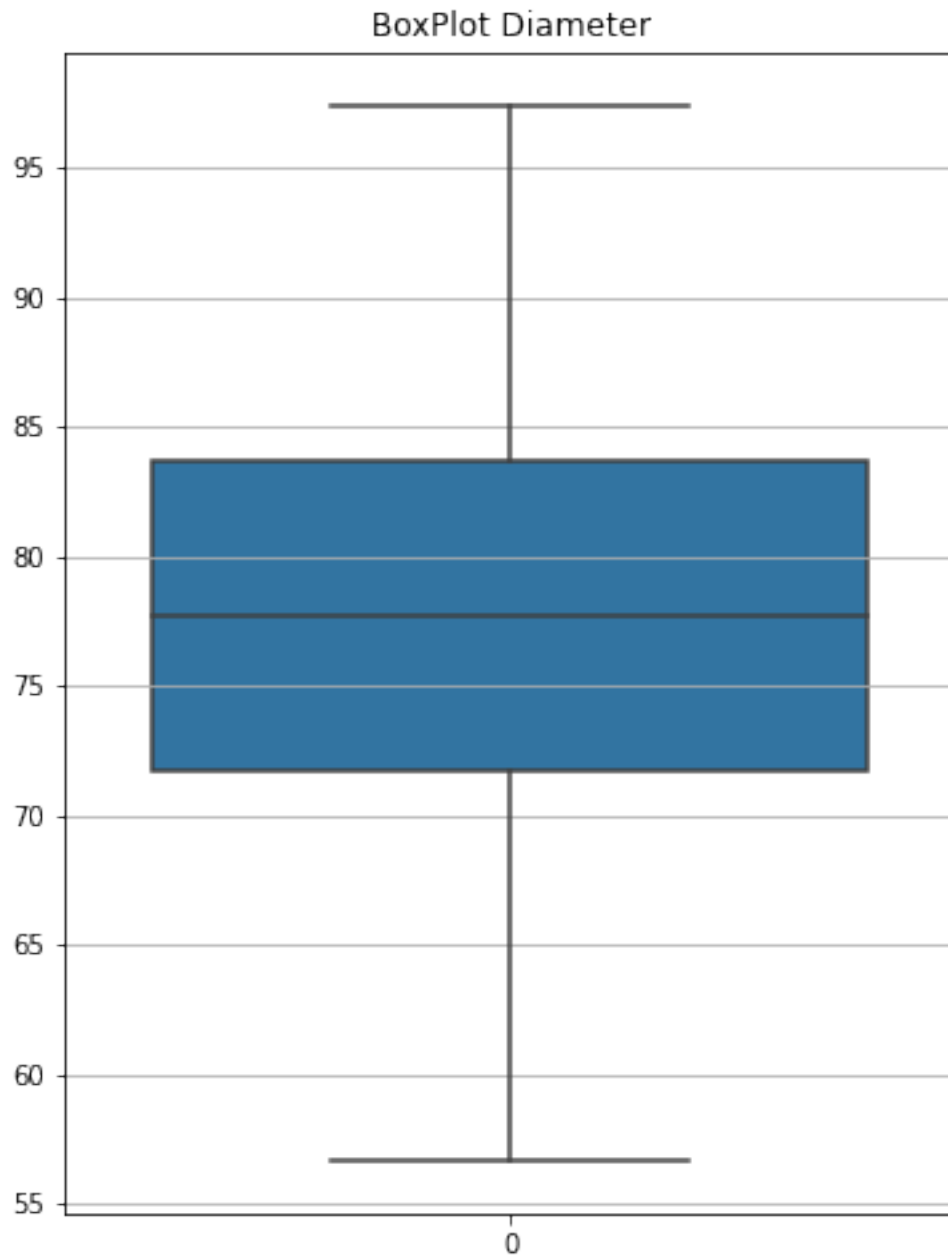
```

```
# Penggambaran Boxplot
BoxPlotDaerah= sns.boxplot(data=Diameter)
BoxPlotDaerah.set_title("BoxPlot Diameter")
```

Proporsi nilai Diameter yang lebih dari 85, adalah tidak sama dengan 15%

1.  $H_0 : p=15$
2.  $H_1 : p \neq 15$
3.  $\alpha = 0.05$
4. Uji Statistik :  $z = (p^{\wedge} - p_0) / \sqrt{p_0 * q_0 / n}$ ,  $\sigma$  diketahui  
Daerah Kritis :  $z > z_{\alpha/2}$  atau  $z < -z_{\alpha/2} : z > 1.96$  atau  $z < -1.96$
5. Komputasi  
 $p^{\wedge} : 0.194$   
 $p_0 : 0.15$   
 $q_0 : 0.85$   
 $n : 500$   
 $p\_value : 0.005869213546929597$   
 $z : 2.755$
6. Test Daerah Kritis  
Tolak  $H_0$  karena nilai uji =  $2.755 > 1.96$  ( $z > z_{\alpha/2}$ )  
Proporsi nilai Diameter yang lebih dari 85 tidak sama dengan 15%

[48]: Text(0.5, 1.0, 'BoxPlot Diameter')



### 7.5 E.Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari 5%

Digunakan uji statistik berupa uji proporsi dengan n besar (binomial didekati normal).

[49]: `print("Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari 5%")`

```
# Langkah 1
H0 = "p=0.05"
print("1. H0 : {}".format(H0))
```

```

# Langkah 2
H1 = "p<0.05"
print("2. H1 : {}".format(H1))

# Langkah 3
alpha = 5e-2
print("3. alpha = {}".format(alpha))

# Langkah 4
z_alpha = round(st.norm.ppf(1-alpha),3)
print("4. Uji Statistik : z=(p^-p0)/sqrt(p0*q0/n), sigma diketahui")
print("    Daerah Kritis : z <-z_alpha : z<{}".format(-1*z_alpha))

# Langkah 5
newKel = [kel for kel in Keliling if kel < 100]
p_hat = len(newKel)/len(Keliling)
p0 = 0.05
q0 = 1-p0
n = len(Keliling)
z = round(Z_testStatistic_bigN(p_hat,p0,q0,n),3)
p_value = st.norm.cdf(z)
print("5. Komputasi")
print("    p_hat : {} \n    p0: {} \n    q0: {} \n    n : {}".format(p_hat,p0,q0,n))
print("    p_value : {} \n    z: {}".format(str(p_value),str(z)))

# Langkah 6
print("6. Test Daerah Kritis")
if (z < -1*z_alpha):
    print("    Tolak H0 karena nilai uji = {}<{} (z<-z_alpha)".
    →format(str(z),str(-1*z_alpha)))
    print("    Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari_
    →5%")
else :
    print("    Terima H0 karena nilai uji = {}>={} (z>=z_alpha)".
    →format(str(z),str(z_alpha)))
    print("    Proporsi nilai Keliling yang kurang dari 100, adalah sama dengan_
    →5%")

# Menggambar Boxplot Keliling

# Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()

# Penggambaran Boxplot
BoxPlotDaerah= sns.boxplot(data=Keliling)

```



```
BoxPlotDaerah.set_title("BoxPlot Keliling")
```

Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari 5%

1.  $H_0 : p=0.05$

2.  $H_1 : p<0.05$

3.  $\alpha = 0.05$

4. Uji Statistik :  $z=(\hat{p}-p_0)/\sqrt{p_0*q_0/n}$ ,  $\sigma$  diketahui

Daerah Kritis :  $z < -z_\alpha : z < -1.645$

5. Komputasi

$\hat{p} : 0.0$

$p_0 : 0.05$

$q_0 : 0.95$

$n : 500$

$p\_value : 1.4487108930250803e-07$

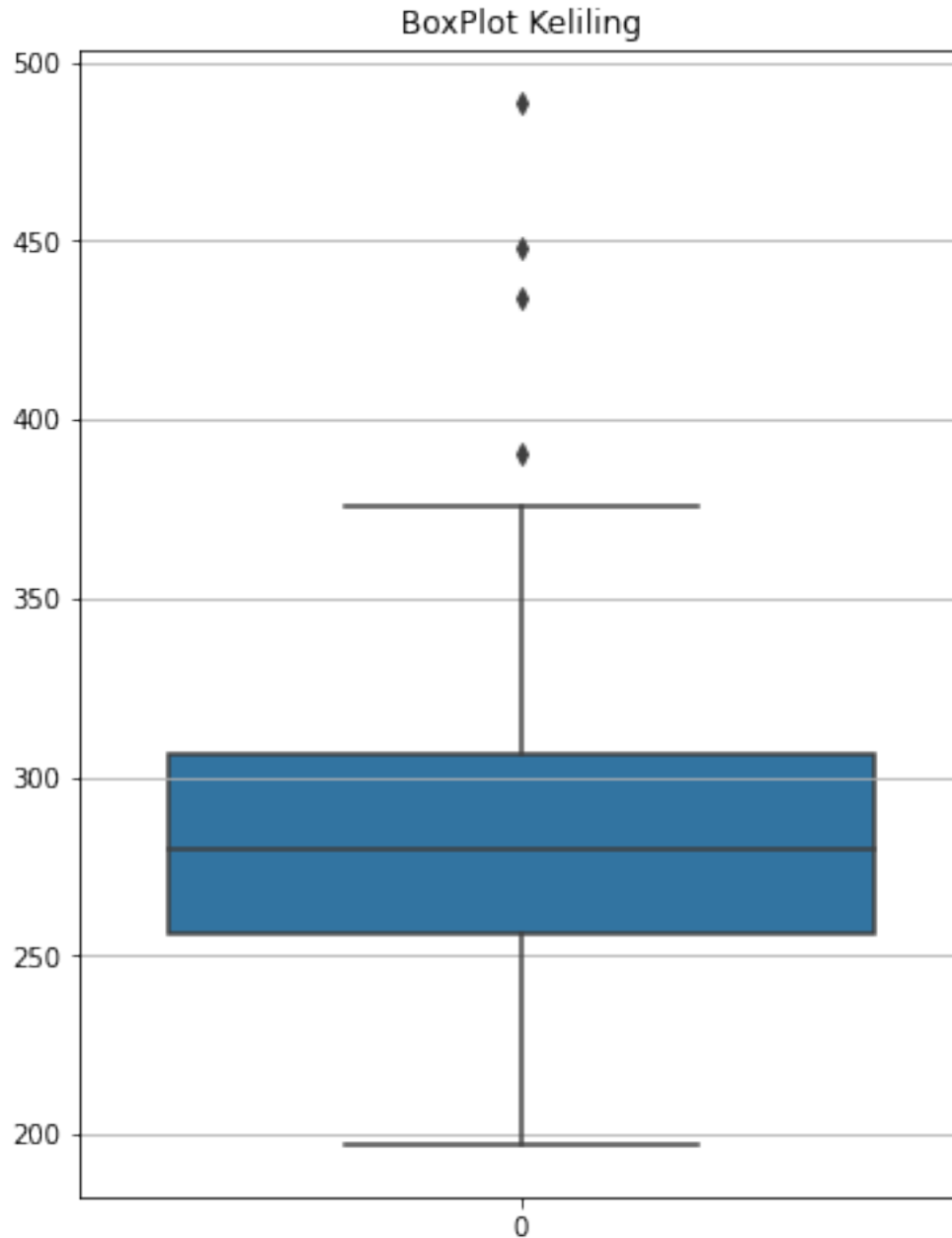
$z : -5.13$

6. Test Daerah Kritis

Tolak  $H_0$  karena nilai uji =  $-5.13 < -1.645$  ( $z < -z_\alpha$ )

Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari 5%

[49]: `Text(0.5, 1.0, 'BoxPlot Keliling')`



## 8 Section 5 - Tes Hipotesis Dua Sampel

Melakukan test hipotesis 2 sampel, dengan menuliskan 6 langkah testing dan menampilkan juga boxplotnya untuk kolom/bagian yang bersesuaian.

```
[50]: def Z_testStatistic_dual(x-1,x-2,d0,σ1_pow2,σ2_pow2,n1,n2):
      return (float)((x-1-x-2)-d0)/np.sqrt(σ1_pow2/n1+σ2_pow2/n2)
      def Z_testStatistic_dualNormal(p^1,p^2,p^,q^,n1,n2):
```

```

    return (float)(p^1-p^2)/np.sqrt(p^*q^(1/n1+1/n2))
def X_testStatistic(n,s, $\sigma$ 0):
    return (float)((n-1)*(s**2))/ $\sigma$ 0**2
def X_testStatisticDual(s1,s2):
    return (float)(s1**2/s2**2)
def SP(n1,n2,s1_pow2,s2_pow2):
    return (float)(np.sqrt(((n1-1)*s1_pow2+(n2-1)*s2_pow2)/(n1+n2-2)))
def T_testStatistic_dual(x^-1,x^-2,d0,sp,n1,n2):
    return (float)((x^-1-x^-2)-d0)/(sp*np.sqrt(1/n1+1/n2))

```

### 8.1 A. Data kolom AreaBulatan dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata kedua bagian tersebut sama?

Asumsi pembagian kolom AreaBulatan menjadi 2 akan menghasilkan dua sampel berbeda dengan tiap sampel berjumlah 250 baris dari populasi sejumlah 500 baris. Karena itu digunakan uji statistik distribusi z untuk dua mean dengan standar deviasi populasi **diketahui**.

```

[51]: print("Data kolom AreaBulatan dibagi 2 sama rata: awal dan akhir kolom. Benarkah
      →rata-rata kedua bagian tersebut sama?")

# Langkah 1
H0 = " $\mu_1=\mu_2$ "
print("1. H0 : {}".format(H0))

# Langkah 2
H1 = " $\mu_1\neq\mu_2 \rightarrow \mu_1-\mu_2\neq 0$ "
print("2. H1 : {}".format(H1))

# Langkah 3
 $\alpha$  = 5e-2
print("3.  $\alpha$  = {}".format( $\alpha$ ))

# Langkah 4
z $\alpha$ _div2 = round(st.norm.ppf(1-( $\alpha$ /2)),3)
print("4. Uji Statistik : z=((x^-1-x^-2)-d0)/sqrt( $\sigma_1^2/n_1+\sigma_2^2/n_2$ ),  $\sigma$  diketahui")
print("   Daerah Kritis : z>z $\alpha$ /2 atau z <-z $\alpha$ /2 : z>{} atau z<{}".
      →format(z $\alpha$ _div2,z $\alpha$ _div2))

# Langkah 5
d0 = 0
AreaBulatan1 = AreaBulatan[:len(AreaBulatan)//2]
x^-1 = AreaBulatan1.mean()
 $\sigma$ 1 = AreaBulatan1.std()
 $\sigma$ 1_pow2 =  $\sigma$ 1**2
n1 = len(AreaBulatan1)
AreaBulatan2 = AreaBulatan[len(AreaBulatan)//2:]
AreaBulatan2.reset_index(inplace=True, drop=True)
x^-2 = AreaBulatan2.mean()

```

```

σ2 = AreaBulatan.std()
σ2_pow2 = σ2**2
n2 = len(AreaBulatan2)
z = round(Z_testStatistic_dual(x-1,x-2,d0,σ1_pow2,σ2_pow2,n1,n2),3)
p_value = 1-abs(st.norm.cdf(z)-st.norm.cdf(-1*z))
print("5. Komputasi")
print("  x-1 : {} \n  σ1: {} \n  σ12: {} \n  n1: {} \n  x-2: {} \n  σ2: {} \n
→ \n  σ22: {} \n  n2: {}".format(x-1,σ1,σ1_pow2,n1,x-2,σ2,σ2_pow2,n2))
print("  d0: {} \n  p_value : {} \n  z: {}".format(d0,str(p_value),str(z)))

# Langkah 6
print("6. Test Daerah Kritis")
if (z > zα_div2 or z < -1*zα_div2):
    if (z > zα_div2):
        print("  Tolak H0 karena nilai uji = {}>{} (z>zα)".format(z,zα_div2))
    else:
        print("  Tolak H0 karena nilai uji = {}<{} (z<zα)".format(z,-1*zα_div2))
    print("  Rata-rata kedua bagian Area Bulatan tidak sama")
else :
    print("  Terima H0 karena nilai uji = {}<{}<{} (-zα<z<zα)".
→format(-1*zα_div2,z,zα_div2))
    print("  Rata-rata kedua bagian Area Bulatan sama")

dfAreaBulatan1 = pd.DataFrame(data=AreaBulatan1.tolist()).assign(AreaBulatanKe=1)
dfAreaBulatan2 = pd.DataFrame(data=AreaBulatan2.tolist()).assign(AreaBulatanKe=2)

combine = pd.concat([dfAreaBulatan1,dfAreaBulatan2],sort=False)
merge = pd.melt(combine, id_vars=['AreaBulatanKe'], var_name=['AreaBulatan'])
merge

# Konfigurasi Boxplot
plt.figure(figsize=(6,8))
plt.grid()
# Penggambaran Boxplot Gabungan
BoxPlotAreaBulatanm= sns.
→boxplot(x="AreaBulatan",y="value",hue="AreaBulatanKe",data=merge)
BoxPlotAreaBulatanm.set_title("BoxPlot AreaBulatan1 dan AreaBulatan2")
BoxPlotAreaBulatanm

```

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

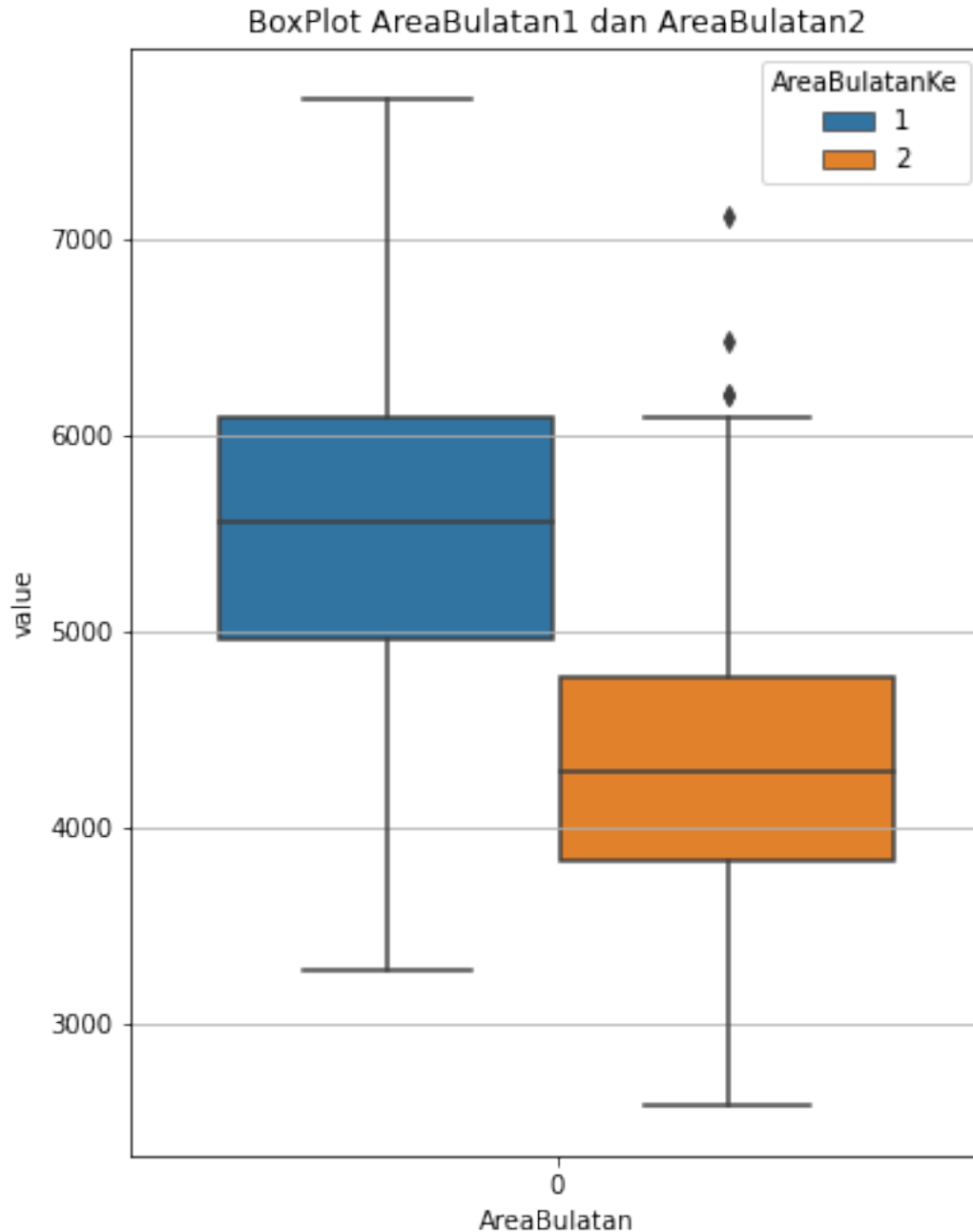
1.  $H_0 : \mu_1 = \mu_2$
2.  $H_1 : \mu_1 \neq \mu_2 \rightarrow \mu_1 - \mu_2 \neq 0$
3.  $\alpha = 0.05$
4. Uji Statistik :  $z = ((x^{-1} - x^{-2}) - d_0) / \sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}$ ,  $\sigma$  diketahui  
Daerah Kritis :  $z > z_{\alpha/2}$  atau  $z < -z_{\alpha/2}$  :  $z > 1.96$  atau  $z < -1.96$
5. Komputasi

```
x-1 : 5549.804
σ1: 1011.6962549701573
σ12: 1023529.3123206415
n1: 250
x-2: 4324.292
σ2: 1011.6962549701573
σ22: 1023529.3123206415
n2: 250
d0: 0
p_value : 0.0
z: 13.543
```

6. Test Daerah Kritis

Tolak H<sub>0</sub> karena nilai uji = 13.543 > 1.96 ( $z > z_{\alpha}$ )  
Rata-rata kedua bagian Area Bulatan tidak sama

```
[51]: <AxesSubplot:title={'center': 'BoxPlot AreaBulatan1 dan AreaBulatan2'},  
      xlabel='AreaBulatan', ylabel='value'>
```



## 8.2 B. Data kolom KadarAir dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata bagian awal lebih besar dari pada bagian akhir sebesar 0.2?

Asumsi pembagian kolom KadarAir menjadi 2 akan menghasilkan dua sampel berbeda dengan tiap sampel berjumlah 250 baris dari populasi sejumlah 500 baris. Karena itu digunakan uji statistik distribusi z untuk dua mean dengan standar deviasi populasi **diketahui**.

```

[52]: print("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?")

# Langkah 1
H0 = " $\mu_1 - \mu_2 = 0.2$ "
print("1. H0 : {}".format(H0))

# Langkah 2
H1 = " $\mu_1 - \mu_2 < 0.2$ "
print("2. H1 : {}".format(H1))

# Langkah 3
 $\alpha$  = 5e-2
print("3.  $\alpha$  = {}".format( $\alpha$ ))

# Langkah 4

z $\alpha$  = round(st.norm.ppf(1- $\alpha$ ),3)
print("4. Uji Statistik :  $z = ((x_1 - x_2) - d_0) / \sqrt{(\sigma_1^2/n_1 + \sigma_2^2/n_2)}$ ,  $\sigma$  diketahui")
print("    Daerah Kritis :  $z < -z_\alpha$  :  $z < {}$ ".format(-1*z $\alpha$ ))

# Langkah 5
d0 = 0.2
KadarAir1 = KadarAir[:len(KadarAir)//2]
x $_1$  = KadarAir1.mean()
 $\sigma_1$  = KadarAir1.std()
 $\sigma_1$ _pow2 =  $\sigma_1$ **2
n1 = len(KadarAir1)
KadarAir2 = KadarAir[len(KadarAir)//2:]
KadarAir2.reset_index(inplace=True, drop=True)
x $_2$  = KadarAir2.mean()
 $\sigma_2$  = KadarAir2.std()
 $\sigma_2$ _pow2 =  $\sigma_2$ **2
n2 = len(KadarAir2)
z = round(Z_testStatistic_dual(x $_1$ ,x $_2$ ,d0, $\sigma_1$ _pow2, $\sigma_2$ _pow2,n1,n2),3)
p_value = round(st.norm.cdf(z),3)
print("5. Komputasi")
print("    x $_1$  : {} \n     $\sigma_1$ : {} \n     $\sigma_1^2$ : {} \n    n1: {} \n    x $_2$ : {} \n     $\sigma_2$ : {}_
      ↳\n     $\sigma_2^2$ : {} \n    n2: {}".format(x $_1$ , $\sigma_1$ , $\sigma_1$ _pow2,n1,x $_2$ , $\sigma_2$ , $\sigma_2$ _pow2,n2))
print("    d0 : {} \n    p_value : {} \n    z: {}".format(d0,str(p_value),str(z)))

# Langkah 6
print("6. Test Daerah Kritis")
if (z < -1*z $\alpha$ ):
    print("    Tolak H0 karena nilai uji = {}<{} (z<-z $\alpha$ )".
      ↳format(str(z),str(-1*z $\alpha$ )))

```

```

    print("    Selisih rata-rata bagian awal KadarAir dengan bagian akhir_
    ↳KadarAir sebesar kurang dari 0.2 ")
else :
    print("    Terima H0 karena nilai uji = {}<={} (z<=zα)".
    ↳format(str(z),str(zα_div2)))
    print("    Rata-rata bagian awal KadarAir lebih besar dari pada bagian akhir_
    ↳KadarAir sebesar 0.2")

dfKadarAir1 = pd.DataFrame(data=KadarAir1.tolist()).assign(KadarAirKe=1)
dfKadarAir2 = pd.DataFrame(data=KadarAir2.tolist()).assign(KadarAirKe=2)

combine = pd.concat([dfKadarAir1,dfKadarAir2],sort=False)
merge = pd.melt(combine, id_vars=['KadarAirKe'], var_name=['KadarAir'])

# Konfigurasi Boxplot
plt.figure(figsize=(10,8))
plt.grid()
# Penggambaran Boxplot Gabungan
BoxPlotKadarAirm= sns.boxplot(x="KadarAir",y="value",hue="KadarAirKe",data=merge)
BoxPlotKadarAirm.set_title("BoxPlot KadarAir1 dan KadarAir2")
BoxPlotKadarAirm

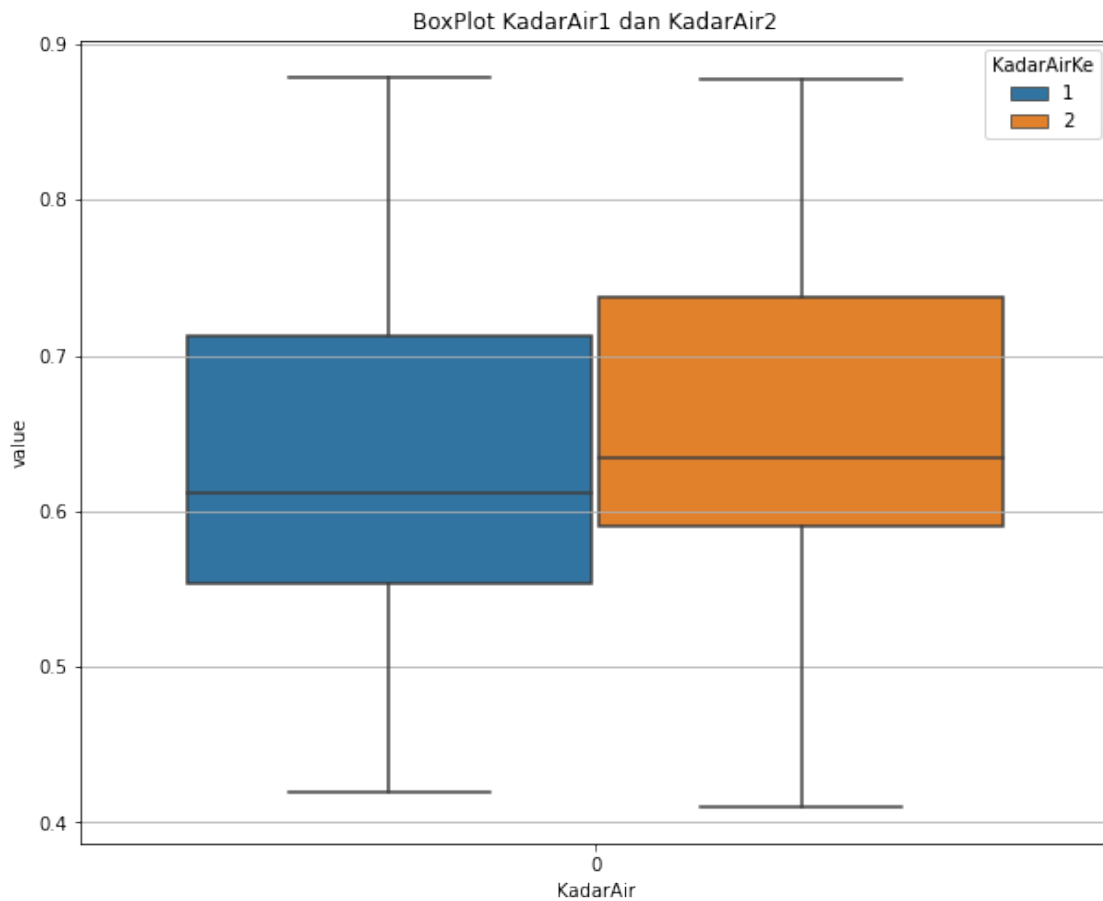
```

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.  $H_0 : \mu_1 - \mu_2 = 0.2$
2.  $H_1 : \mu_1 - \mu_2 < 0.2$
3.  $\alpha = 0.05$
4. Uji Statistik :  $z = ((\bar{x}_1 - \bar{x}_2) - d_0) / \sqrt{(\sigma_1^2/n_1 + \sigma_2^2/n_2)}$ ,  $\sigma$  diketahui  
Daerah Kritis :  $z < -z_\alpha : z < -1.645$
5. Komputasi
  - $\bar{x}_1$  : 0.63574344072
  - $\sigma_1$ : 0.09436709809379533
  - $\sigma_1^2$ : 0.00890514920264399
  - $n_1$ : 250
  - $\bar{x}_2$ : 0.660999903076
  - $\sigma_2$ : 0.09436709809379533
  - $\sigma_2^2$ : 0.00890514920264399
  - $n_2$ : 250
  - $d_0$  : 0.2
  - $p\_value$  : 0.0
  - $z$ : -26.688
6. Test Daerah Kritis
  - Tolak  $H_0$  karena nilai uji = -26.688 < -1.645 ( $z < -z_\alpha$ )
  - Selisih rata-rata bagian awal KadarAir dengan bagian akhir KadarAir sebesar kurang dari 0.2



```
[52]: <AxesSubplot:title={'center':'BoxPlot KadarAir1 dan KadarAir2'},
      xlabel='KadarAir', ylabel='value'>
```



### 8.3 C. Rata-rata 20 baris pertama kolom Bulatan sama dengan 20 baris terakhirnya?

Asumsi pembagian kolom Bulatan menjadi dua yaitu 20 baris pertama dan 20 baris terakhir akan menghasilkan dua sampel berbeda dari populasi yang sama sejumlah 500 baris. Karena itu digunakan uji statistik distribusi z untuk dua mean dengan standar deviasi populasi **diketahui**.

```
[53]: print("Rata-rata 20 baris pertama kolom Bulatan sama dengan 20 baris terakhirnya?
      ↪")
      # Langkah 1
      H0 = " $\mu_1 = \mu_2$ "
      print("1. H0 : {}".format(H0))

      # Langkah 2
      H1 = " $\mu_1 \neq \mu_2 \rightarrow \mu_1 - \mu_2 \neq 0$ "
      print("2. H1 : {}".format(H1))
```

```

# Langkah 3
 $\alpha$  = 5e-2
print("3.  $\alpha$  = {}".format( $\alpha$ ))

# Langkah 4
z $\alpha$ _div2 = round(st.norm.ppf(1-( $\alpha$ /2)),3)
print("4. Uji Statistik :  $z=((x^{-1}-x^{-2})-d0)/\sqrt{(\sigma_1^2/n1+\sigma_2^2/n2)}$ ,  $\sigma$  diketahui")
print("    Daerah Kritis :  $z > z\alpha/2$  atau  $z < -z\alpha/2$  :  $z > \{\}$  atau  $z < \{\}$ ".
     $\rightarrow$ format(z $\alpha$ _div2,z $\alpha$ _div2))

# Langkah 5
d0 = 0
Bulatan1 = Bulatan[:20]
x $^{-1}$  = Bulatan1.mean()
 $\sigma_1$  = Bulatan1.std()
 $\sigma_1$ _pow2 =  $\sigma_1$ **2
n1 = len(Bulatan1)
Bulatan2 = Bulatan[len(AreaBulatan)-20:]
Bulatan2.reset_index(inplace=True, drop=True)
x $^{-2}$  = Bulatan2.mean()
 $\sigma_2$  = Bulatan2.std()
 $\sigma_2$ _pow2 =  $\sigma_2$ **2
n2 = len(Bulatan2)
z = round(Z_testStatistic_dual(x $^{-1}$ ,x $^{-2}$ ,d0, $\sigma_1$ _pow2, $\sigma_2$ _pow2,n1,n2),3)
p_value = 1-abs(st.norm.cdf(z)-st.norm.cdf(-1*z))
print("5. Komputasi")
print("    x $^{-1}$  : {} \n     $\sigma_1$  : {} \n     $\sigma_1^2$ : {} \n    n1 : {} \n    x $^{-2}$  : {} \n     $\sigma_2$  :
     $\rightarrow$  {} \n     $\sigma_2^2$ : {} \n    n2 : {}".format(x $^{-1}$ , $\sigma_1$ , $\sigma_1$ _pow2,n1,x $^{-2}$ , $\sigma_2$ , $\sigma_2$ _pow2,n2))
print("    p_value : {} \n    z: {}".format(str(p_value),str(z)))

# Langkah 6
print("6. Test Daerah Kritis")
if (z > z $\alpha$ _div2 or z < -1*z $\alpha$ _div2):
    if (z > z $\alpha$ _div2):
        print("    Tolak H0 karena nilai uji = {}>{} (z>z $\alpha$ )".format(z,z $\alpha$ _div2))
    else:
        print("    Tolak H0 karena nilai uji = {}<{} (z<-z $\alpha$ )".
             $\rightarrow$ format(z,-1*z $\alpha$ _div2))
        print("    Rata-rata 20 baris pertama kolom Bulatan tidak sama dengan 20_
             $\rightarrow$ baris terakhirnya")
else :
    print("    Terima H0 karena nilai uji = {}<{}<{} (-z $\alpha$ <z<z $\alpha$ )".
         $\rightarrow$ format(-1*z $\alpha$ _div2,z,z $\alpha$ _div2))
    print("    Rata-rata 20 baris pertama kolom Bulatan sama dengan 20 baris_
         $\rightarrow$ terakhirnya")

```

```

dfBulatan1 = pd.DataFrame(data=Bulatan1.tolist()).assign(BulatanKe=1)
dfBulatan2 = pd.DataFrame(data=Bulatan2.tolist()).assign(BulatanKe=2)

combine = pd.concat([dfBulatan1,dfBulatan2],sort=False)
merge = pd.melt(combine, id_vars=['BulatanKe'], var_name=['Bulatan'])

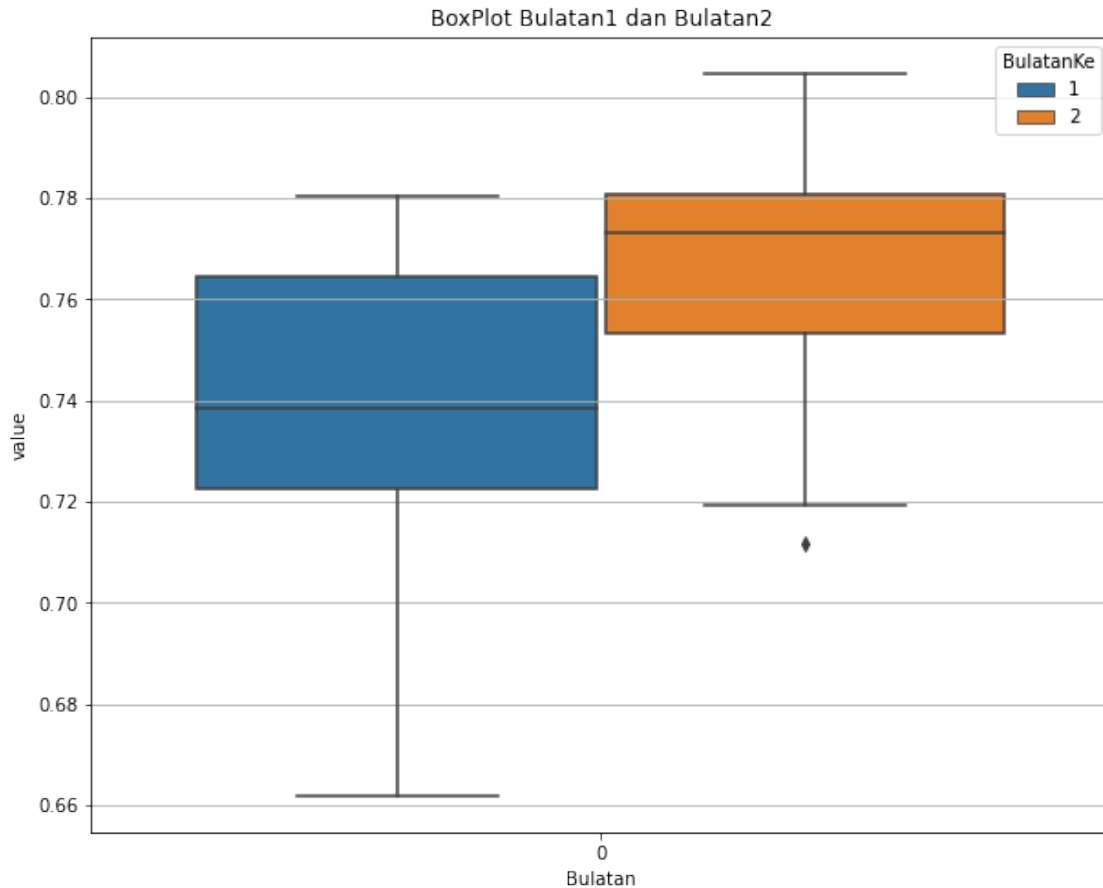
# Konfigurasi Boxplot
plt.figure(figsize=(10,8))
plt.grid()
# Penggambaran Boxplot Gabungan
BoxPlotBulatanm= sns.boxplot(x="Bulatan",y="value",hue="BulatanKe",data=merge)
BoxPlotBulatanm.set_title("BoxPlot Bulatan1 dan Bulatan2")
BoxPlotBulatanm

```

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

1.  $H_0 : \mu_1 = \mu_2$
2.  $H_1 : \mu_1 \neq \mu_2 \rightarrow \mu_1 - \mu_2 \neq 0$
3.  $\alpha = 0.05$
4. Uji Statistik :  $z = ((x^{-1} - x^{-2}) - d_0) / \sqrt{(\sigma_1^2/n_1 + \sigma_2^2/n_2)}$ ,  $\sigma$  diketahui  
Daerah Kritis :  $z > z_{\alpha/2}$  atau  $z < -z_{\alpha/2}$  :  $z > 1.96$  atau  $z < -1.96$
5. Komputasi  
 $x^{-1}$  : 0.73753535525  
 $\sigma_1$  : 0.06170246078673261  
 $\sigma_1^2$ : 0.0038071936671382756  
 $n_1$  : 20  
 $x^{-2}$  : 0.767322437  
 $\sigma_2$  : 0.06170246078673261  
 $\sigma_2^2$ : 0.0038071936671382756  
 $n_2$  : 20  
 $p\_value$  : 0.12676101137143414  
 $z$ : -1.527
6. Test Daerah Kritis  
Terima  $H_0$  karena nilai uji =  $-1.96 < -1.527 < 1.96$  ( $-z_{\alpha} < z < z_{\alpha}$ )  
Rata-rata 20 baris pertama kolom Bulatan sama dengan 20 baris terakhirnya

[53]: <AxesSubplot:title={'center': 'BoxPlot Bulatan1 dan Bulatan2'}, xlabel='Bulatan', ylabel='value'>



#### 8.4 D. Proporsi nilai bagian awal Ransum yang lebih dari 2, adalah lebih besar daripada, proporsi nilai yang sama di bagian akhir Ransum?

Digunakan tes dua proporsi

```
[54]: print("Proporsi bagian awal Ransum yang lebih dari 2 lebih besar daripada,
→proporsi nilai yang sama di bagian akhir Ransum")
# Langkah 1
H0 = "p1=p2 -> p1-p2=0"
print("1. H0 : {}".format(H0))

# Langkah 2
H1 = "p1>p2 -> p1-p2>0"
print("2. H1 : {}".format(H1))

# Langkah 3
α = 5e-2
print("3. α = {}".format(α))
```



```

# Konfigurasi Boxplot
plt.figure(figsize=(10,8))
plt.grid()

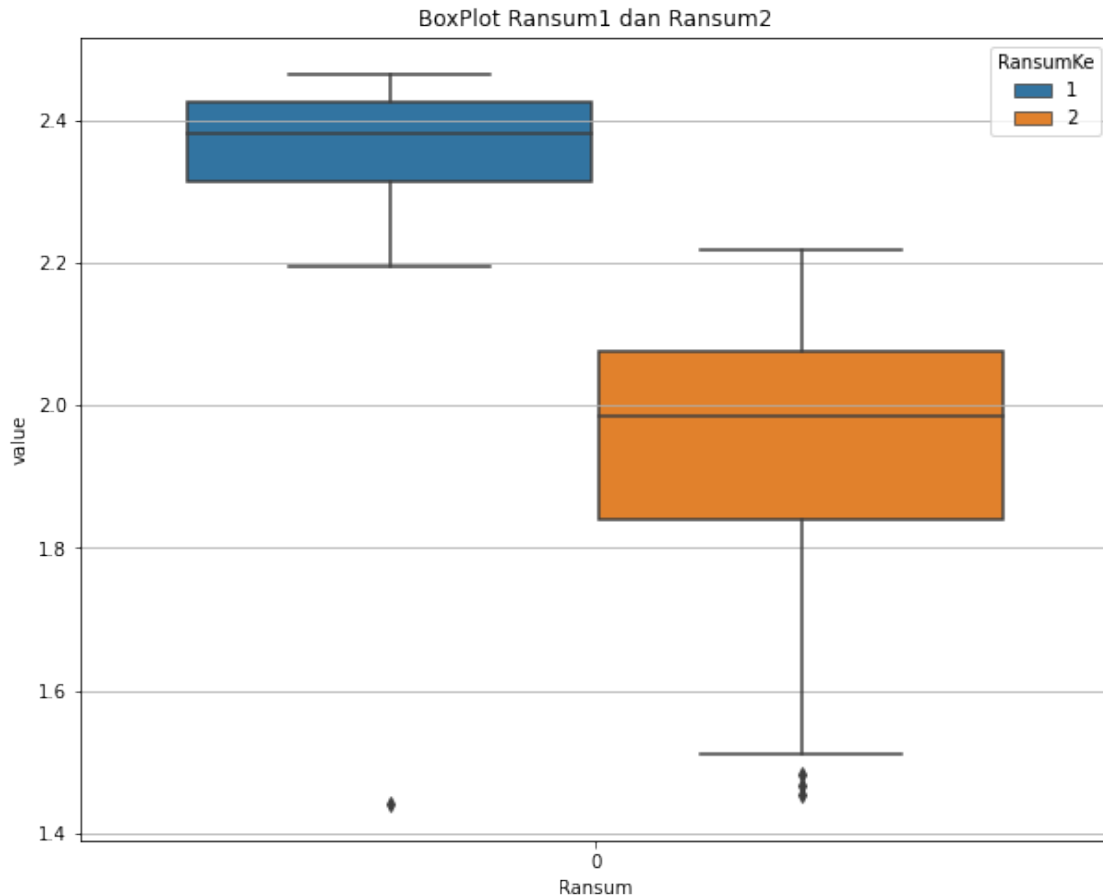
# Penggambaran Boxplot Gabungan
BoxPlotRansumm= sns.boxplot(x="Ransum",y="value",hue="RansumKe",data=merge)
BoxPlotRansumm.set_title("BoxPlot Ransum1 dan Ransum2")
BoxPlotRansumm

```

Proporsi bagian awal Ransum yang lebih dari 2 lebih besar daripada, proporsi nilai yang sama di bagian akhir Ransum

1.  $H_0 : p_1=p_2 \rightarrow p_1-p_2=0$
2.  $H_1 : p_1>p_2 \rightarrow p_1-p_2>0$
3.  $\alpha = 0.05$
4. Uji Statistik :  $z=(p^1-p^2)/\sqrt{p^*q^*(1/n_1+1/n_2)}$ ,  $\sigma$  diketahui  
Daerah Kritis :  $z>z\alpha : z>1.645$
5. Komputasi
  - n1 : 250
  - x1: 249
  - $p^1$ : 0.996
  - n2: 250
  - x2: 116
  - $p^2$ : 0.464
  - $p^*$ : 0.73
  - $q^*$ : 0.27
  - p\_value : 0.0
  - z: 13.397
6. Test Daerah Kritis
  - Tolak  $H_0$  karena nilai uji =  $13.397>1.645$  ( $z>z\alpha$ )
  - Proporsi bagian awal Ransum yang lebih dari 2 lebih besar daripada proporsi nilai yang sama di bagian akhir Ransum

[54]: <AxesSubplot:title={'center':'BoxPlot Ransum1 dan Ransum2'}, xlabel='Ransum', ylabel='value'>



### 8.5 E. Bagian awal kolom Diameter memiliki variansi yang sama dengan bagian akhirnya?

Digunakan tes variansi. Pada kasus ini digunakan alfa sebesar 0.01 agar  $\alpha/2$  yang didapatkan 0.005 dan ada di tabel di buku walpole.

```
[55]: print("Rata-rata 20 baris pertama kolom Bulatan sama dengan 20 baris terakhirnya?
      ↪")
# Langkah 1
H0 = " $\sigma_1^2 = \sigma_2^2$ "
print("1. H0 : {}".format(H0))

# Langkah 2
H1 = " $\sigma_1^2 \neq \sigma_2^2 \rightarrow \sigma_1^2 - \sigma_2^2 \neq 0$ "
print("2. H1 : {}".format(H1))

# Langkah 3
 $\alpha$  = 1e-1
print("3.  $\alpha$  = {}".format( $\alpha$ ))
```

```

# Langkah 4
Diameter1 = Diameter[:len(Diameter)//2]
Diameter2 = Diameter[len(Diameter)//2:]
Diameter2.reset_index(inplace=True, drop=True)
n1 = len(Diameter1)
n2 = len(Diameter2)
v1 = n1-1
v2 = n2-1
f $\alpha$ _div2 = round(st.f.ppf(q=1- $\alpha$ /2, dfn=v1, dfd=v2),3)
f1_min_ $\alpha$ div2 = round((1.0/round(st.f.ppf(q=1- $\alpha$ /2, dfn=v2, dfd=v1),3)),3)
print("4. Uji Statistik : z=((x-1-x-2)-d0)/np.sqrt( $\sigma_1^2/n_1+\sigma_2^2/n_2$ ),  $\sigma$  diketahui")
print("    Daerah Kritis : f>f $\alpha$ /2(v1,v2) atau f <f1- $\alpha$ /2(v1,v2) : f>{} atau f<{}".
    →format(f $\alpha$ _div2,f1_min_ $\alpha$ div2))

# Langkah 5
s1 = Diameter1.std()
s2 = Diameter2.std()
f = round(X_testStatisticDual(s1,s2),3)

print("5. Komputasi")
print("    n1 : {} \n     $\sigma^2_1$ : {} \n    v1: {} \n    n2: {} \n     $\sigma^2_2$ : {} \n    v2: {}".
    →format(n1,s1**2,v1,n2,s2**2,v2))

# # Langkah 6
print("6. Test Daerah Kritis")
if (f > f $\alpha$ _div2 or f < f1_min_ $\alpha$ div2):
    if (f > f $\alpha$ _div2):
        print("    Tolak H0 karena nilai uji = {}>{} (f > f $\alpha$ /2)".
            →format(f,f $\alpha$ _div2))
    else:
        print("    Tolak H0 karena nilai uji = {}<{} (f < f1- $\alpha$ /2)".
            →format(f,f1_min_ $\alpha$ div2))
        print("    Bagian awal kolom Diameter memiliki variansi yang tidak sama,
            →dengan bagian akhirnya")
else :
    print("    Terima H0 karena nilai uji = {}<{}<{} (-z $\alpha$ <z<z $\alpha$ )".
        →format(f1_min_ $\alpha$ div2,f,f $\alpha$ _div2))
    print("    Bagian awal kolom Diameter memiliki variansi yang sama dengan,
        →bagian akhirnya")

# Menggambar BoxPlot AreaBulatan1 dan AreaBulatan2

# Merubah ke dalam bentuk dataframe
dfDiameter1 = pd.DataFrame(data=Diameter1.tolist()).assign(DiameterKe=1)

```



```

dfDiameter2 = pd.DataFrame(data=Diameter2.tolist()).assign(DiameterKe=2)

# Mengkombinasikan dataframe
combine = pd.concat([dfDiameter1,dfDiameter2],sort=False)
merge = pd.melt(combine, id_vars=['DiameterKe'], var_name=['Diameter'])

# Konfigurasi Boxplot
plt.figure(figsize=(10,8))
plt.grid()

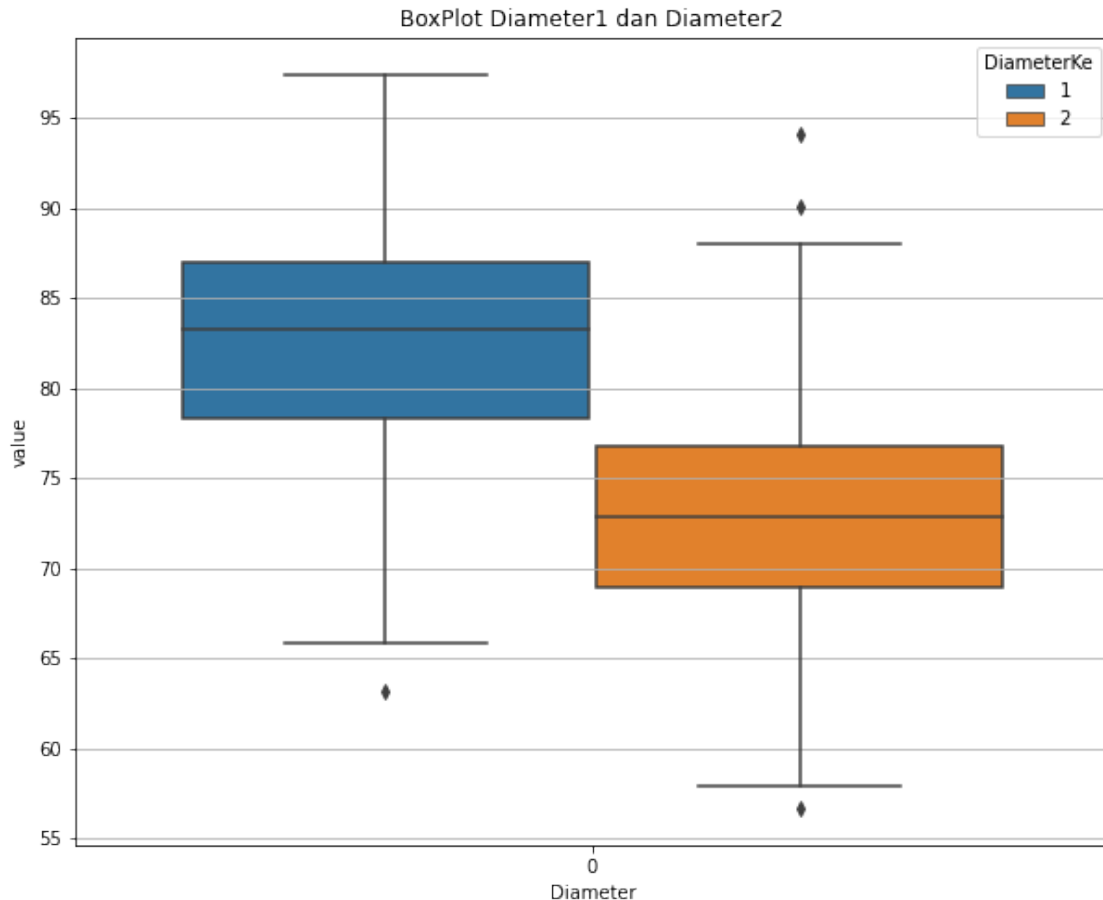
# Penggambaran Boxplot Gabungan
BoxPlotDiameterm= sns.boxplot(x="Diameter",y="value",hue="DiameterKe",data=merge)
BoxPlotDiameterm.set_title("BoxPlot Diameter1 dan Diameter2")
BoxPlotDiameterm

```

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

1.  $H_0 : \sigma_1^2 = \sigma_2^2$
2.  $H_1 : \sigma_1^2 \neq \sigma_2^2 \rightarrow \sigma_1^2 - \sigma_2^2 \neq 0$
3.  $\alpha = 0.1$
4. Uji Statistik :  $z = ((x^{-1} - x^{-2}) - d_0) / \text{np.sqrt}(\sigma_1^2/n_1 + \sigma_2^2/n_2)$ ,  $\sigma$  diketahui  
Daerah Kritis :  $f > f_{\alpha/2}(v_1, v_2)$  atau  $f < f_{1-\alpha/2}(v_1, v_2)$  :  $f > 1.232$  atau  $f < 0.812$
5. Komputasi  
 $n_1$  : 250  
 $\sigma^2_1$ : 43.168980426322435  
 $v_1$ : 249  
 $n_2$ : 250  
 $\sigma^2_2$ : 39.8282643880772  
 $v_2$ : 249
6. Test Daerah Kritis  
Terima  $H_0$  karena nilai uji =  $0.812 < 1.084 < 1.232$  ( $-z_{\alpha} < z < z_{\alpha}$ )  
Bagian awal kolom Diameter memiliki variansi yang sama dengan bagian akhirnya

[55]: <AxesSubplot:title={'center': 'BoxPlot Diameter1 dan Diameter2'},  
xlabel='Diameter', ylabel='value'>



## 9 Section 6 - Tes Korelasi

Mentukan apakah setiap kolom non-target berkorelasi dengan kolom target, dengan menggambarkan juga scatter plot nya. Tes korelasi menggunakan correlation test. Daftar kolom non-target adalah sebagai berikut : 1. Daerah 2. SumbuUtama 3. SumbuKecil 4. Keunikan 5. AreaBulatan 6. Diameter 7. KadarAir 8. Keliling 9. Bulatan 10. Ransum

Sedangkan kolom target hanya kelas.

Metode yang digunakan sebagai berikut : 1. Melakukan tes korelasi dengan metode Pearson dengan  $\alpha = 0.05$  -  $H_0 : p = 0$  (Tidak ada korelasi diantara kolom non-target dengan kolom target) -  $H_1 : p \neq 0$  (Ada korelasi antara kolom non-target dan kolom target)

- Jika terdapat korelasi diantara kolom non-target dan kolom target maka cari berapa koefisien korelasi diantara kedua kolom tersebut. Beberapa kesimpulan yang dapat diambil sebagai berikut :

- Koefisien Korelasi = 1 (Strong Positive Correlation)
- $0 < \text{Koefisien Korelasi} < 1$  (Positive Correlation)
- Koefisien Korelasi = 0 (No Correlation)

- $-1 < \text{Koefisien Korelasi} < 0$  (Negative Correlation)
- Koefisien Korelasi = -1 (Strong Negative Correlation)

3. Tampilkan Scatter Plot diantara dua kolom tersebut

```
[56]: # Deklarasi dataframe kolom kelas dan nilai alpha
```

```
Kelas = df["Kelas"]
alpha = 0.05
```

### 9.0.1 1. Tes Korelasi Kolom Daerah dengan Kolom Kelas

```
[57]: # Tes Korelasi
stat, p = st.pearsonr(Daerah, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
    ↳target")
else:
    print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
    ↳non-target dan kolom target")
```

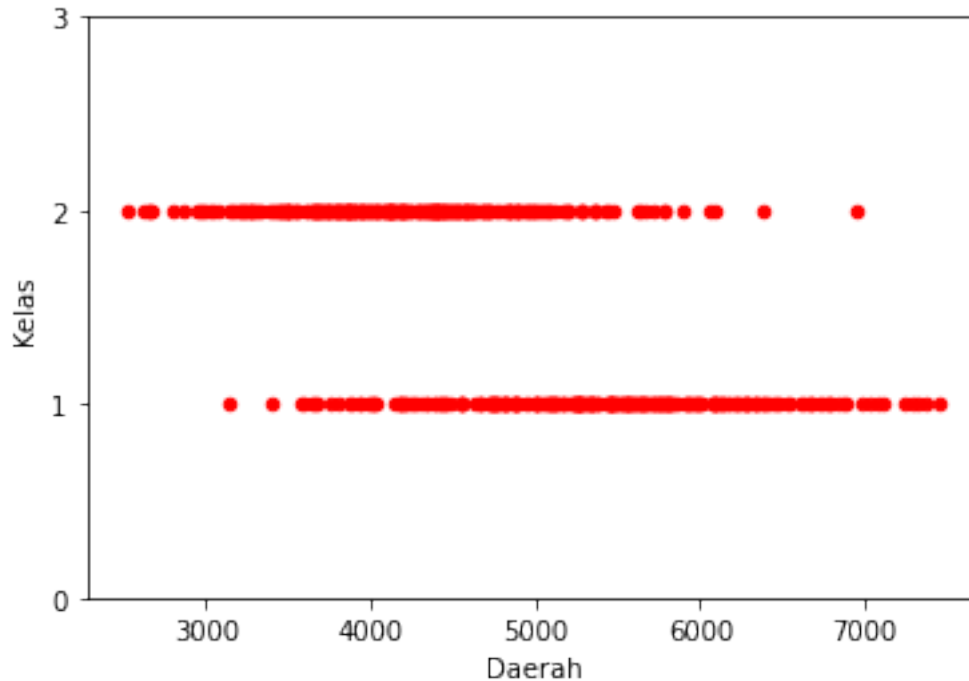
H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

```
[58]: # Koefisien Korelasi
Daerah.corr(Kelas, method='pearson')
```

```
[58]: -0.6027466517416662
```

```
[59]: # Scatter Plot
df.plot(kind='scatter', x='Daerah', y='Kelas', yticks=[0.0, 1.0, 2.0, 3.0],
    ↳color='red')
```

```
[59]: <AxesSubplot: xlabel='Daerah', ylabel='Kelas'>
```



**Kesimpulan** Terdapat korelasi diantara kolom Daerah (non-target) dan kolom Kelas (target) yaitu Negative Correlation. Sehingga nilai daerah yang lebih besar akan dikategorikan pada kelas yang lebih kecil atau sebaliknya.

## 9.0.2 2. Tes Korelasi Kolom SumbuUtama dengan Kolom Kelas

```
[60]: # Tes Korelasi
stat, p = st.pearsonr(SumbuUtama, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
    ↳target")
else:
    print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
    ↳non-target dan kolom target")
```

H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

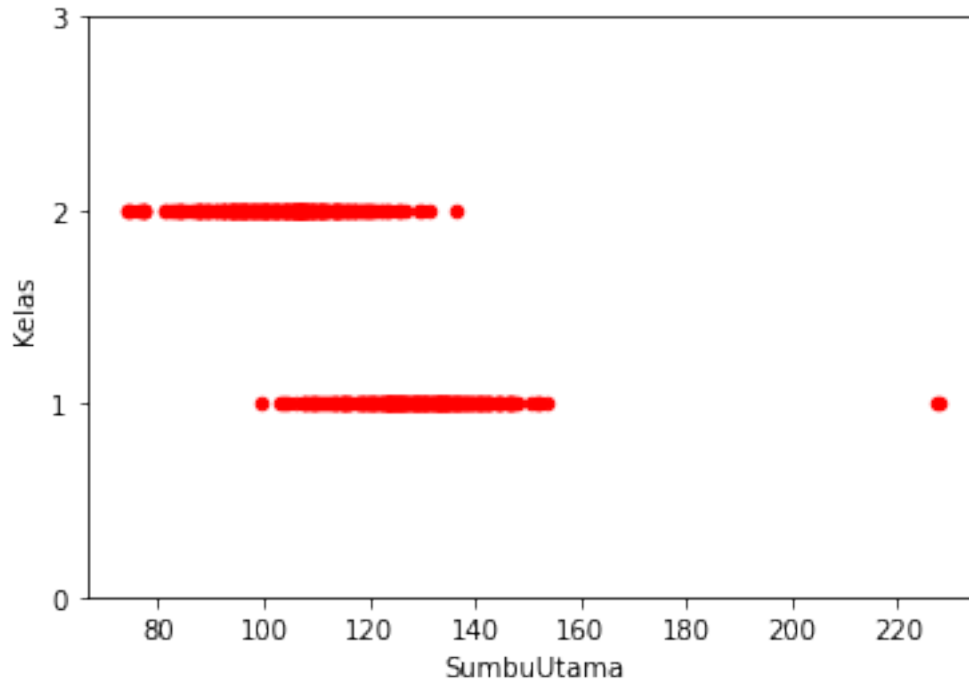
```
[61]: # Koefisien Korelasi
SumbuUtama.corr(Kelas, method='pearson')
```

```
[61]: -0.7130906104204593
```

```
[62]: # Scatter Plot
```

```
df.plot(kind='scatter', x='SumbuUtama', y='Kelas', yticks=[0.0, 1.0, 2.0, 3.0],  
        color='red')
```

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



**Kesimpulan** Terdapat korelasi diantara kolom SumbuUtama (non-target) dan kolom Kelas (target) yaitu Negative Correlation. Sehingga nilai SumbuUtama yang lebih besar akan dikategorikan pada kelas yang lebih kecil atau sebaliknya.

### 9.0.3 3. Tes Korelasi Kolom SumbuKecil dengan Kolom Kelas

```
[63]: # Tes Korelasi
stat, p = st.pearsonr(SumbuKecil, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
    →target")
else:
    print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
    →non-target dan kolom target")
```

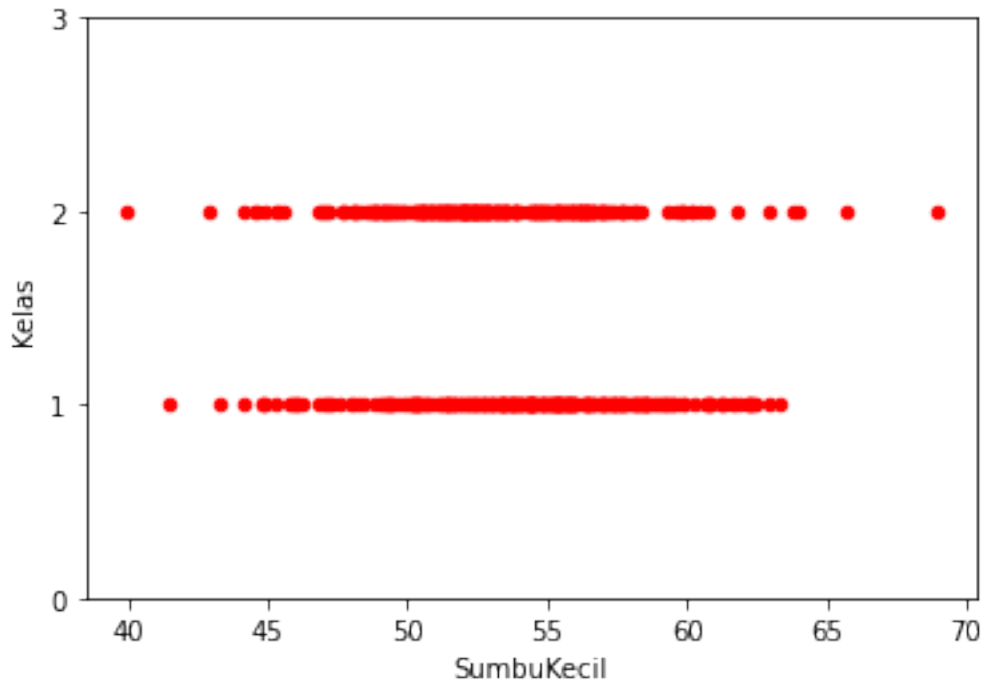
H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

```
[64]: # Koefisien Korelasi
SumbuKecil.corr(Kelas, method='pearson')
```

```
[64]: -0.15297517335535027
```

```
[65]: # Scatter Plot
df.plot(kind='scatter', x='SumbuKecil', y='Kelas', yticks=[0.0, 1.0, 2.0, 3.0],
        color='red')
```

```
[65]: <AxesSubplot: xlabel='SumbuKecil', ylabel='Kelas'>
```



**Kesimpulan** Terdapat korelasi diantara kolom SumbuKecil (non-target) dan kolom Kelas (target) yaitu Negative Correlation. Sehingga nilai SumbuKecil yang lebih besar akan dikategorikan pada kelas yang lebih kecil atau sebaliknya.

#### 9.0.4 4. Tes Korelasi Kolom Keunikan dengan Kolom Kelas

```
[66]: # Tes Korelasi
stat, p = st.pearsonr(Keunikan, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
        target")
else:
    print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
        non-target dan kolom target")
```

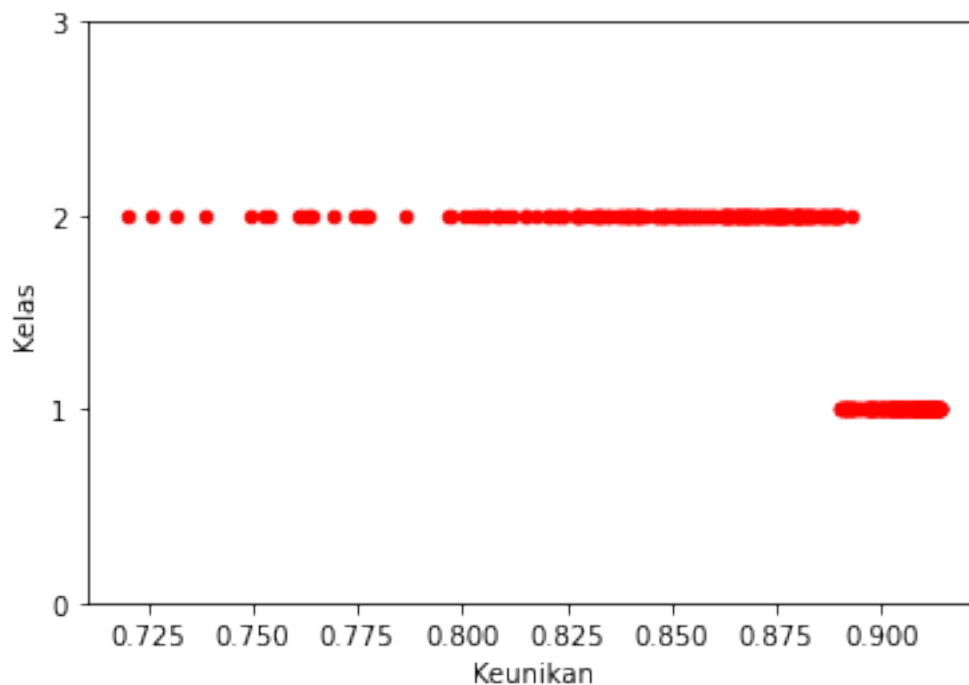
H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

```
[67]: # Koefisien Korelasi
Keunikan.corr(Kelas, method='pearson')
```

```
[67]: -0.7304563686511922
```

```
[68]: # Scatter Plot
df.plot(kind='scatter', x='Keunikan', y='Kelas', yticks=[0.0, 1.0, 2.0, 3.0],
        color='red')
```

```
[68]: <AxesSubplot: xlabel='Keunikan', ylabel='Kelas'>
```



**Kesimpulan** Terdapat korelasi diantara kolom Keunikan (non-target) dan kolom Kelas (target) yaitu Negative Correlation. Sehingga nilai Keunikan yang lebih besar akan dikategorikan pada kelas yang lebih kecil atau sebaliknya.

#### 9.0.5 5. Tes Korelasi Kolom AreaBulatan dengan Kolom Kelas

```
[69]: # Tes Korelasi
stat, p = st.pearsonr(AreaBulatan, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
        target")
else:
```

```
print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
→non-target dan kolom target")
```

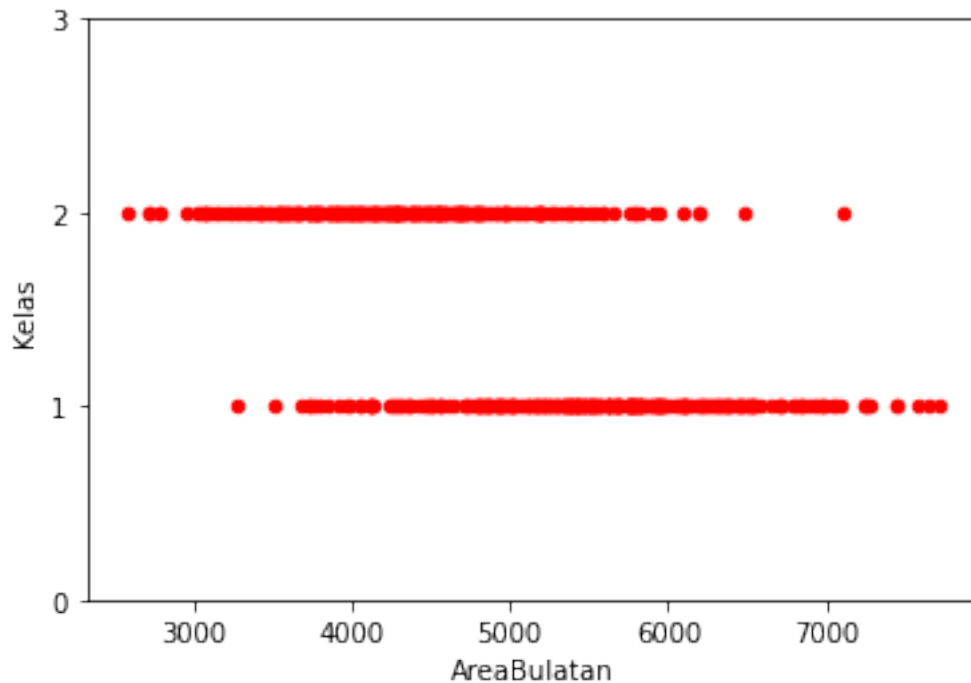
H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

```
[70]: # Koefisien Korelasi
AreaBulatan.corr(Kelas, method='pearson')
```

```
[70]: -0.6073125434153751
```

```
[71]: # Scatter Plot
df.plot(kind='scatter', x='AreaBulatan', y='Kelas', yticks=[0.0, 1.0, 2.0, 3.0],
→color='red')
```

```
[71]: <AxesSubplot: xlabel='AreaBulatan', ylabel='Kelas'>
```



**Kesimpulan** Terdapat korelasi diantara kolom AreaBulatan (non-target) dan kolom Kelas (target) yaitu Negative Correlation. Sehingga nilai AreaBulatan yang lebih besar akan dikategorikan pada kelas yang lebih kecil atau sebaliknya.



### 9.0.6 6. Tes Korelasi Kolom Diameter dengan Kolom Kelas

```
[72]: # Tes Korelasi
stat, p = st.pearsonr(Diameter, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
    ↳target")
else:
    print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
    ↳non-target dan kolom target")
```

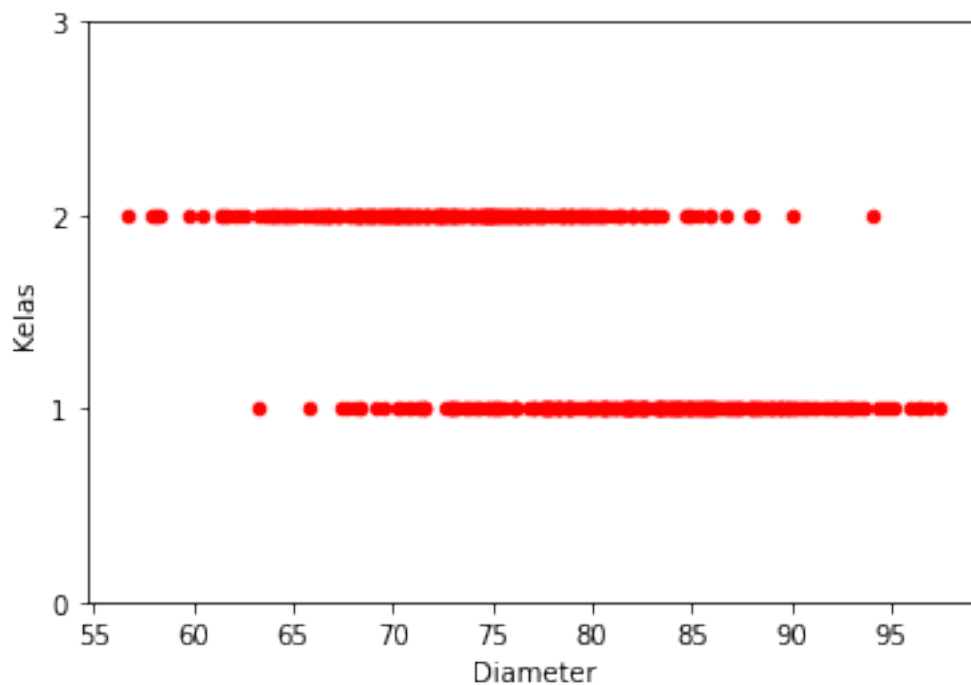
H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

```
[73]: # Koefisien Korelasi
Diameter.corr(Kelas, method='pearson')
```

```
[73]: -0.6025356896618813
```

```
[74]: # Scatter Plot
df.plot(kind='scatter', x='Diameter', y='Kelas', yticks=[0.0, 1.0, 2.0, 3.0],
    ↳color='red')
```

```
[74]: <AxesSubplot:xlabel='Diameter', ylabel='Kelas'>
```



**Kesimpulan** Terdapat korelasi diantara kolom Diameter (non-target) dan kolom Kelas (target) yaitu Negative Correlation. Sehingga nilai Diameter yang lebih besar akan dikategorikan pada kelas yang lebih kecil atau sebaliknya.

#### 9.0.7 7. Tes Korelasi Kolom KadarAir dengan Kolom Kelas

```
[75]: # Tes Korelasi
stat, p = st.pearsonr(KadarAir, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
    ↳target")
else:
    print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
    ↳non-target dan kolom target")
```

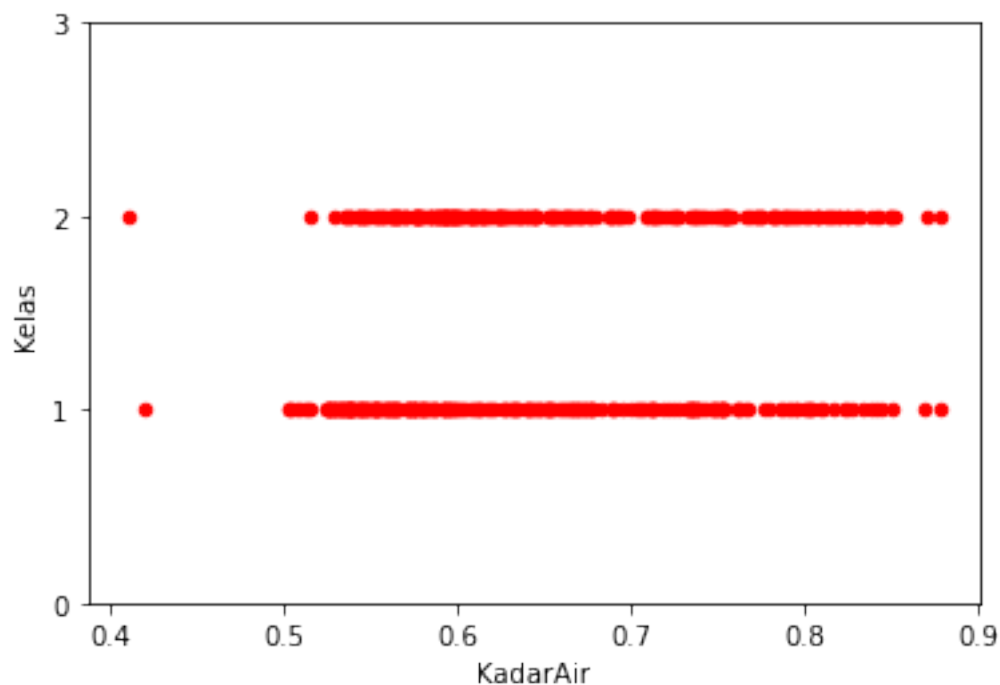
H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

```
[76]: # Koefisien Korelasi
KadarAir.corr(Kelas, method='pearson')
```

```
[76]: 0.13434422605727642
```

```
[77]: # Scatter Plot
df.plot(kind='scatter', x='KadarAir', y='Kelas',yticks=[0.0, 1.0, 2.0, 3.0],
    ↳color='red')
```

```
[77]: <AxesSubplot:xlabel='KadarAir', ylabel='Kelas'>
```



**Kesimpulan** Terdapat korelasi diantara kolom KadarAir (non-target) dan kolom Kelas (target) yaitu Positive Correlation. Sehingga nilai KadarAir yang lebih besar akan dikategorikan pada kelas yang lebih besar atau sebaliknya.

#### 9.0.8 8. Tes Korelasi Kolom Keliling dengan Kolom Kelas

```
[78]: # Tes Korelasi
stat, p = st.pearsonr(Keliling, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
    ↳target")
else:
    print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
    ↳non-target dan kolom target")
```

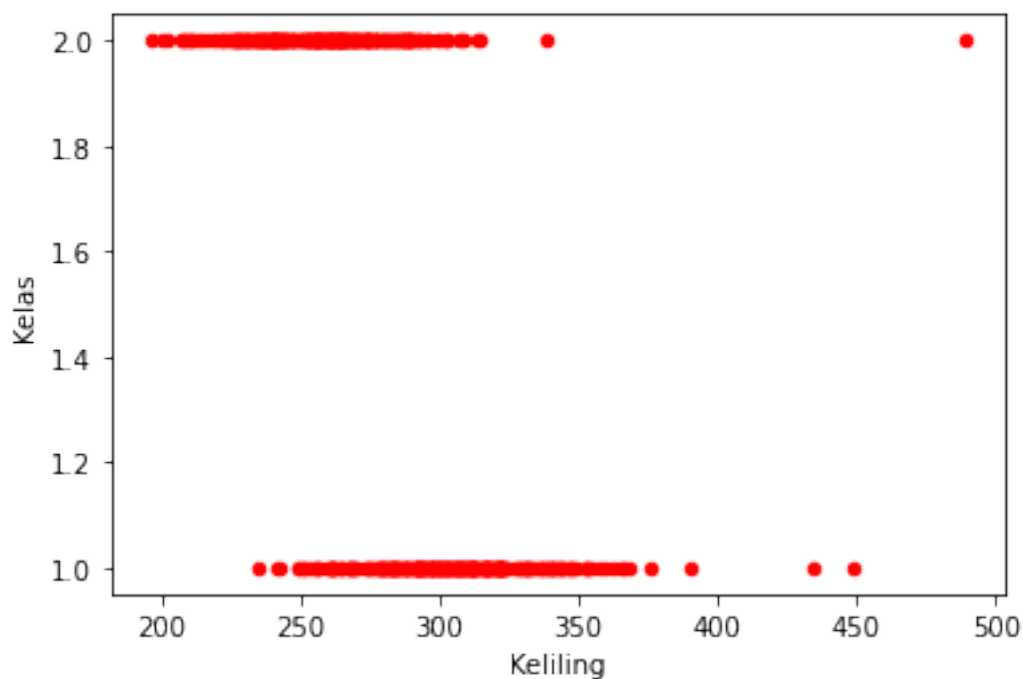
H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

```
[79]: # Koefisien Korelasi
Keliling.corr(Kelas, method='pearson')
```

```
[79]: -0.6348607454756858
```

```
[80]: # Scatter Plot
df.plot(kind='scatter', x='Keliling', y='Kelas', color='red')
```

```
[80]: <AxesSubplot:xlabel='Keliling', ylabel='Kelas'>
```



**Kesimpulan** Terdapat korelasi diantara kolom Keliling (non-target) dan kolom Kelas (target) yaitu Negative Correlation. Sehingga nilai Keliling yang lebih besar akan dikategorikan pada kelas yang lebih kecil atau sebaliknya.

### 9.0.9 9. Tes Korelasi Kolom Bulatan dengan Kolom Kelas

```
[81]: # Tes Korelasi
stat, p = st.pearsonr(Bulatan, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
    ↳target")
else:
    print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
    ↳non-target dan kolom target")
```

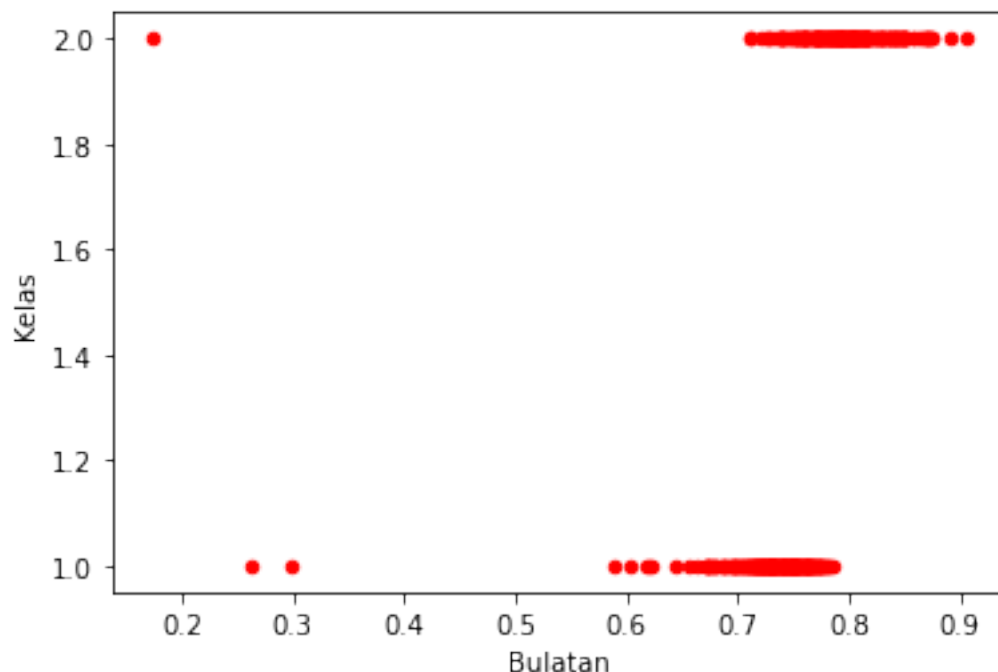
H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

```
[82]: # Koefisien Korelasi
Bulatan.corr(Kelas, method='pearson')
```

```
[82]: 0.5450045317240076
```

```
[83]: # Scatter Plot
df.plot(kind='scatter', x='Bulatan', y='Kelas', color='red')
```

```
[83]: <AxesSubplot:xlabel='Bulatan', ylabel='Kelas'>
```



**Kesimpulan** Terdapat korelasi diantara kolom Daerah (non-target) dan kolom Kelas (target) yaitu Positive Correlation. Sehingga nilai Bulatan yang lebih besar akan dikategorikan pada kelas yang lebih besar atau sebaliknya.

#### 9.0.10 10. Tes Korelasi Kolom Ransum dengan Kolom Kelas

```
[84]: # Tes Korelasi
stat, p = st.pearsonr(Ransum, Kelas)
if p < alpha:
    print("H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom_
    ↳target")
else:
    print("H0 tidak bisa ditolak, sehingga tidak ada korelasi antara kolom_
    ↳non-target dan kolom target")
```

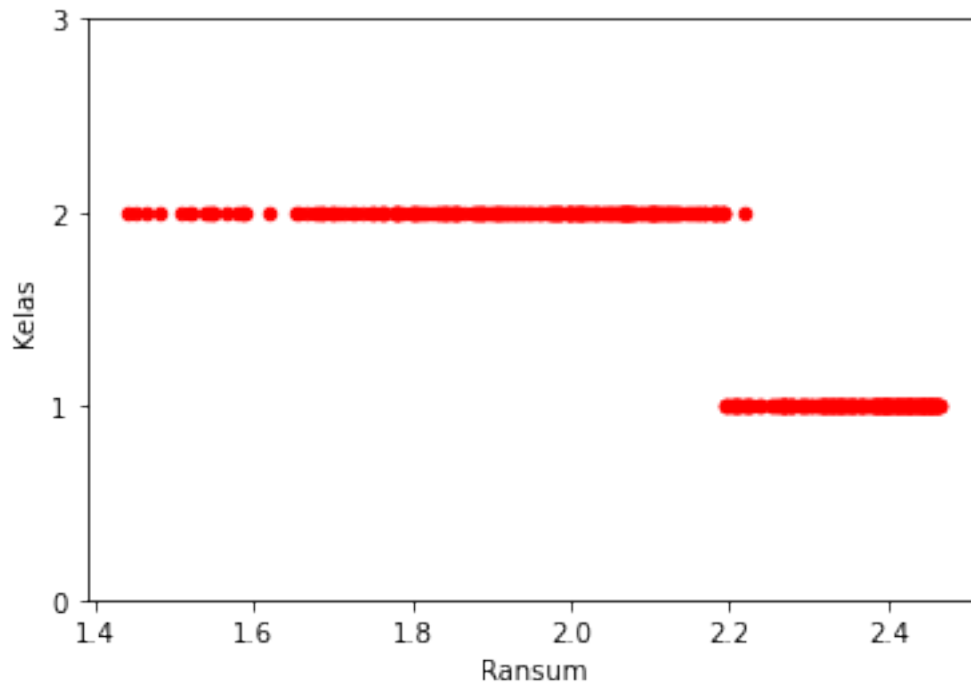
H0 ditolak, sehingga ada korelasi antara kolom non-target dan kolom target

```
[85]: # Koefisien Korelasi
Ransum.corr(Kelas, method='pearson')
```

```
[85]: -0.8399038681287493
```

```
[86]: # Scatter Plot
df.plot(kind='scatter', x='Ransum', y='Kelas', yticks=[0.0, 1.0, 2.0, 3.0],
    ↳color='red')
```

```
[86]: <AxesSubplot:xlabel='Ransum', ylabel='Kelas'>
```



**Kesimpulan** Terdapat korelasi diantara kolom Ransum (non-target) dan kolom Kelas (target) yaitu Negative Correlation. Sehingga nilai Ransum yang lebih besar akan dikategorikan pada kelas yang lebih kecil atau sebaliknya.