02_T1_IF2220_13519081

April 11, 2021

1 Tugas Besar I Probabilitas dan Statistika Semester II Tahun Akademik 2020/2021

---

[5]:
```
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as s
import seaborn as sns
alpha = 0.05
colnames = ['id', 'daerah', 'sumbu utama', 'sumbu kecil', 'keunikan', 'area␣
 ↪bulatan', 'diameter', 'kadar air', 'keliling', 'bulatan', 'ransum', 'kelas']
url = 'https://drive.google.com/file/d/1z0EyNb2c5PD-ECNgsS_6f7KV6bm2ya10/view?
 ↪usp=sharing' # copy of original dataset, in personal drive
path = 'https://drive.google.com/uc?export=download&id='+url.split('/')[-2]
g = pd.read_csv(path, header=None)
g.columns = colnames
gandum = g[g.columns[1:11]] #gstripped
```

##1. *Descriptive Statistics*

[6]:
```
gandum.describe()
```

[6]:

	daerah	sumbu utama	sumbu kecil	keunikan	area bulatan	\
count	500.000000	500.000000	500.000000	500.000000	500.000000	
mean	4801.246000	116.045171	53.715246	0.878764	4937.048000	
std	986.395491	18.282626	4.071075	0.036586	1011.696255	
min	2522.000000	74.133114	39.906517	0.719916	2579.000000	
25%	4042.750000	104.116098	51.193576	0.863676	4170.250000	
50%	4735.000000	115.405140	53.731199	0.890045	4857.000000	
75%	5495.500000	129.046792	56.325158	0.907578	5654.250000	
max	7453.000000	227.928583	68.977700	0.914001	7720.000000	

	diameter	kadar air	keliling	bulatan	ransum
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	77.771158	0.648372	281.479722	0.761737	2.150915
std	8.056867	0.094367	37.335402	0.061702	0.249767
min	56.666658	0.409927	197.015000	0.174590	1.440796

```
25%     71.745308    0.572632  255.883000    0.731991    1.983939
50%     77.645277    0.626117  280.045500    0.761288    2.193599
75%     83.648598    0.726633  306.062500    0.796361    2.381612
max     97.413830    0.878899  488.837000    0.904748    2.464809
```

Mean

[7]:
```
gandum.mean() #mean
```

[7]:
```
daerah          4801.246000
sumbu utama      116.045171
sumbu kecil       53.715246
keunikan           0.878764
area bulatan    4937.048000
diameter          77.771158
kadar air          0.648372
keliling         281.479722
bulatan            0.761737
ransum             2.150915
dtype: float64
```

Median

[8]:
```
gandum.median() #median
```

[8]:
```
daerah          4735.000000
sumbu utama      115.405140
sumbu kecil       53.731199
keunikan           0.890045
area bulatan    4857.000000
diameter          77.645277
kadar air          0.626117
keliling         280.045500
bulatan            0.761288
ransum             2.193599
dtype: float64
```

Modus

[9]:
```
for i in colnames:
  if i != 'id' and i != 'kelas':
    mod = gandum[i].mode()
    print()
    print("Modus " + i + ": ", end="")
    for j in range(len(mod)):
      if j != len(mod) -1:
        print(mod[j], end= ", ")
      else:
        print(mod[j])
```

555, 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

Modus area bulatan: 3802, 4913

Modus diameter: 71.29356396, 78.83325579, 84.75622403, 88.00634154

Modus kadar air: 0.735849057, 0.824404762

Modus keliling: 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

Modus bulatan: 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, 0.736669733, 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

Modus ransum: 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

[10]:
```
gandum.std() #standar deviasi
```

[10]:
```
daerah          986.395491
sumbu utama      18.282626
sumbu kecil       4.071075
keunikan          0.036586
area bulatan    1011.696255
diameter          8.056867
kadar air         0.094367
keliling         37.335402
bulatan           0.061702
ransum            0.249767
dtype: float64
```

Variansi

[11]:
```
gandum.var() # variansi
```

[11]:
```
daerah          9.729761e+05
sumbu utama     3.342544e+02
sumbu kecil     1.657365e+01
keunikan        1.338528e-03
area bulatan    1.023529e+06
diameter        6.491311e+01
kadar air       8.905149e-03
keliling        1.393932e+03
bulatan         3.807194e-03
ransum          6.238350e-02
dtype: float64
```

Range

[12]:
```
gandum.max() - gandum.min() #range
```

[12]:
```
daerah          4931.000000
sumbu utama      153.795469
sumbu kecil       29.071182
keunikan           0.194085
area bulatan    5141.000000
diameter          40.747172
kadar air          0.468972
keliling         291.822000
bulatan            0.730158
ransum             1.024013
dtype: float64
```

Min

[13]:
```
gandum.min() #min
```

[13]:
```
daerah          2522.000000
sumbu utama       74.133114
sumbu kecil       39.906517
keunikan           0.719916
area bulatan    2579.000000
diameter          56.666658
kadar air          0.409927
keliling         197.015000
bulatan            0.174590
ransum             1.440796
dtype: float64
```

Max

[14]:
```
gandum.max() #max
```

[14]:
```
daerah          7453.000000
sumbu utama      227.928583
sumbu kecil       68.977700
keunikan           0.914001
area bulatan    7720.000000
diameter          97.413830
kadar air          0.878899
keliling         488.837000
bulatan            0.904748
ransum             2.464809
dtype: float64
```

Kuartil

[15]:
```
gandum.quantile([0.25,0.5,0.75]) #kuartil (quantil 0.25, 0.5, 0.75)
```

[15]:

	daerah	sumbu utama	sumbu kecil	keunikan	area bulatan	diameter	\
0.25	4042.75	104.116098	51.193576	0.863676	4170.25	71.745308	
0.50	4735.00	115.405140	53.731199	0.890045	4857.00	77.645277	
0.75	5495.50	129.046792	56.325158	0.907578	5654.25	83.648598	

	kadar air	keliling	bulatan	ransum
0.25	0.572632	255.8830	0.731991	1.983939
0.50	0.626117	280.0455	0.761288	2.193599
0.75	0.726633	306.0625	0.796361	2.381612

IQR

[16]:
```
gandum.quantile(0.75) - gandum.quantile(0.25) #IQR
```

[16]:
```
daerah          1452.750000
sumbu utama       24.930694
sumbu kecil        5.131582
keunikan           0.043902
area bulatan    1484.000000
diameter          11.903290
kadar air          0.154001
keliling          50.179500
bulatan            0.064370
ransum             0.397673
dtype: float64
```

Skewness

[17]:
```
gandum.skew()
```

[17]:
```
daerah          0.238144
sumbu utama     0.761529
sumbu kecil    -0.010828
```

```
keunikan       -1.623472
area bulatan    0.257560
diameter        0.002725
kadar air       0.493661
keliling        0.733627
bulatan        -3.599237
ransum         -0.658188
dtype: float64
```

Kurtosis

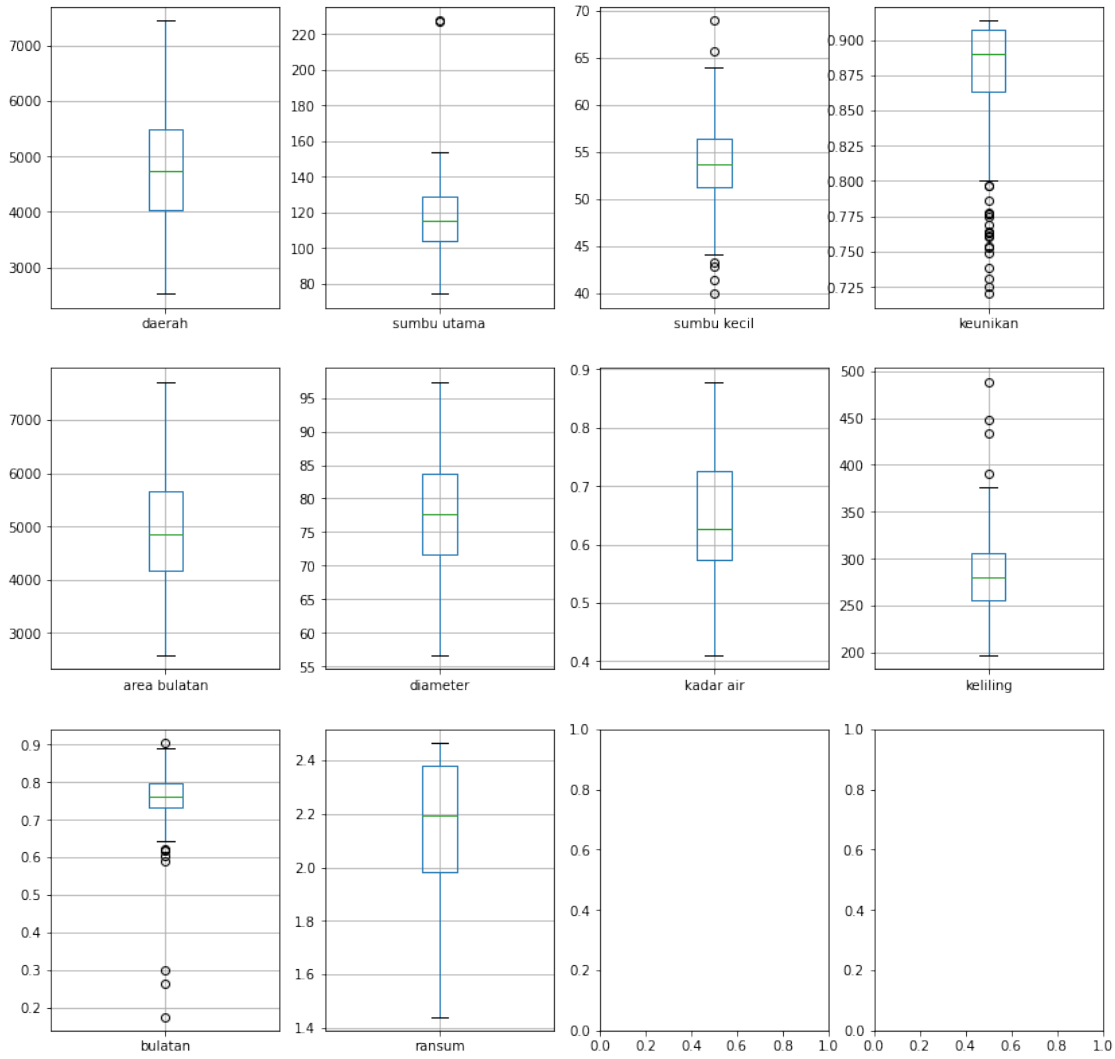
[18]:
```
gandum.kurtosis()
```

[18]:
```
daerah          -0.434631
sumbu utama      4.330534
sumbu kecil      0.475568
keunikan         2.917256
area bulatan    -0.409685
diameter        -0.466455
kadar air       -0.740326
keliling         2.272685
bulatan         29.975096
ransum          -0.428656
dtype: float64
```

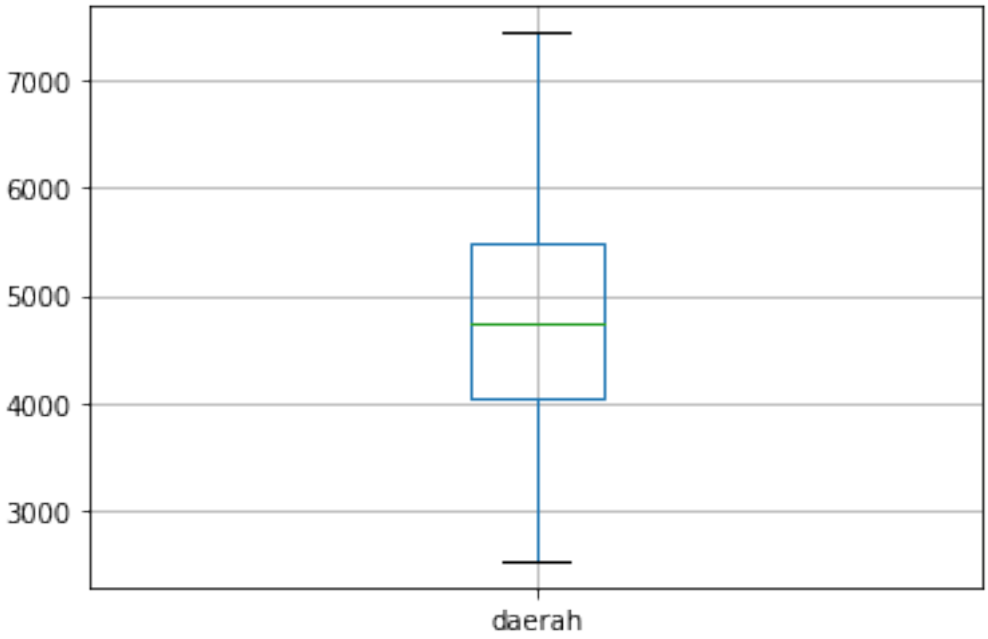
1.1 2. Visualisasi

Boxplot

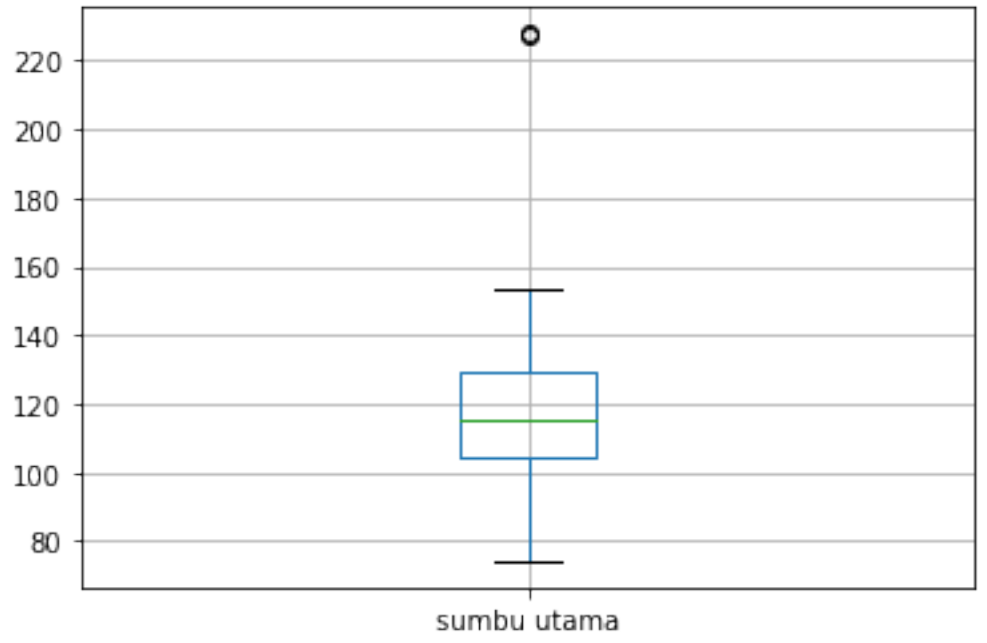
[19]:
```
fig, axes = plt.subplots(nrows = 3, ncols=4, figsize=(14,14))

row = 0
for i, col in enumerate(gandum.columns):
  gandum.boxplot(column=col, ax=axes[row, i%4])
  if i % 4 == 3:
    row += 1
plt.show()
```



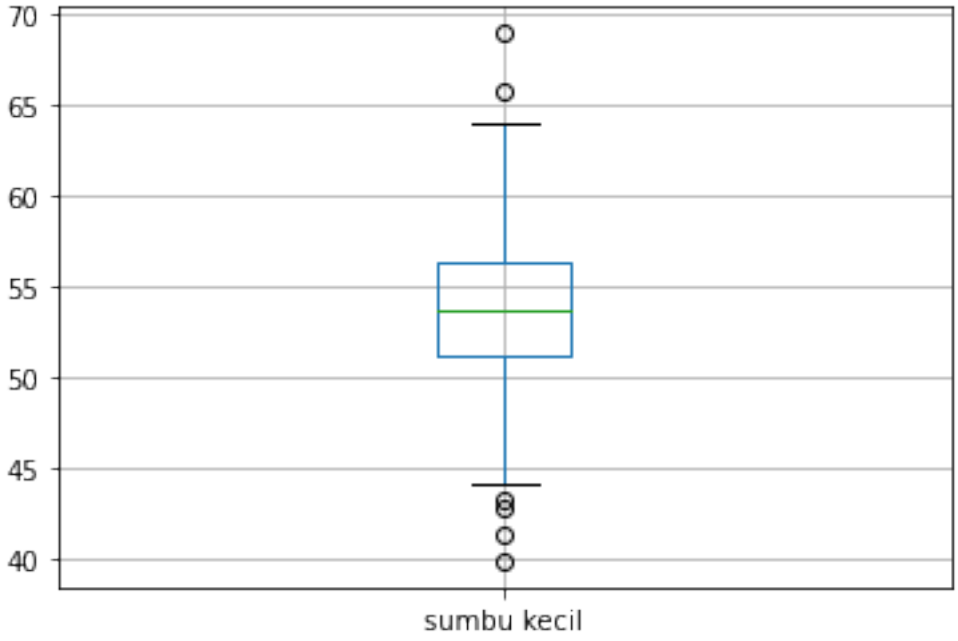
[20]:
```
df = pd.DataFrame(data = gandum, columns = ['daerah'])
boxplot = df.boxplot(column = ['daerah'])
```



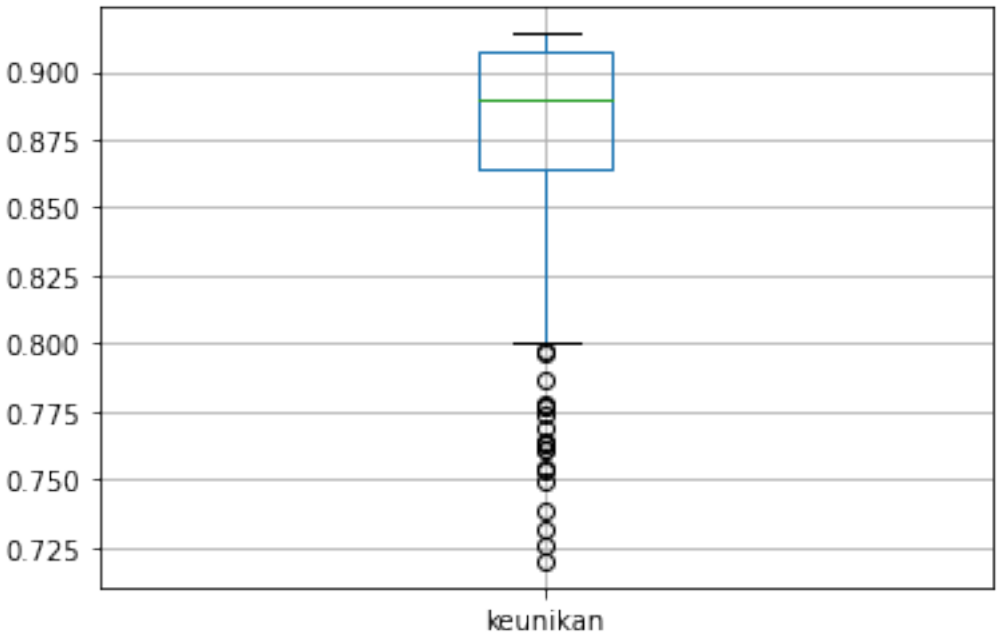
[21]:
```
df = pd.DataFrame(data = gandum, columns = ['sumbu utama'])
boxplot = df.boxplot(column = ['sumbu utama'])
```



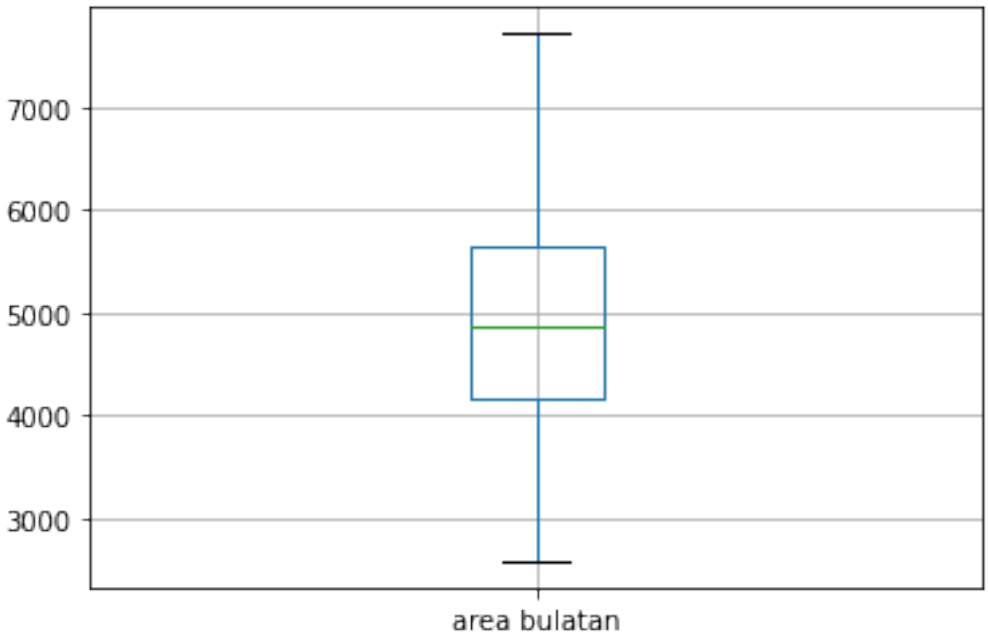
[22]:
```
df = pd.DataFrame(data = gandum, columns = ['sumbu kecil'])
boxplot = df.boxplot(column = ['sumbu kecil'])
```



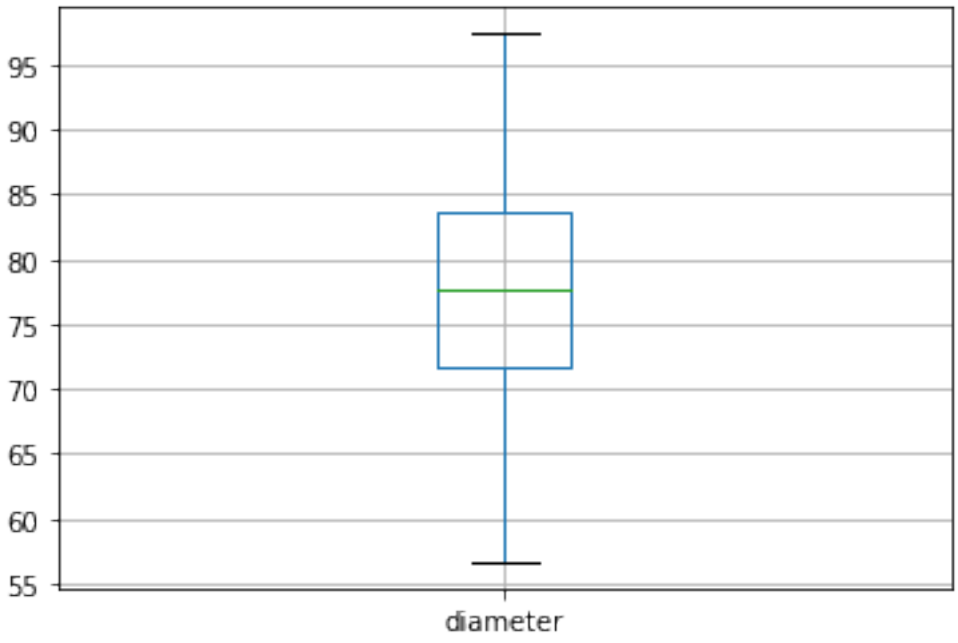
[23]:
```
df = pd.DataFrame(data = gandum, columns = ['keunikan'])
boxplot = df.boxplot(column = ['keunikan'])
```



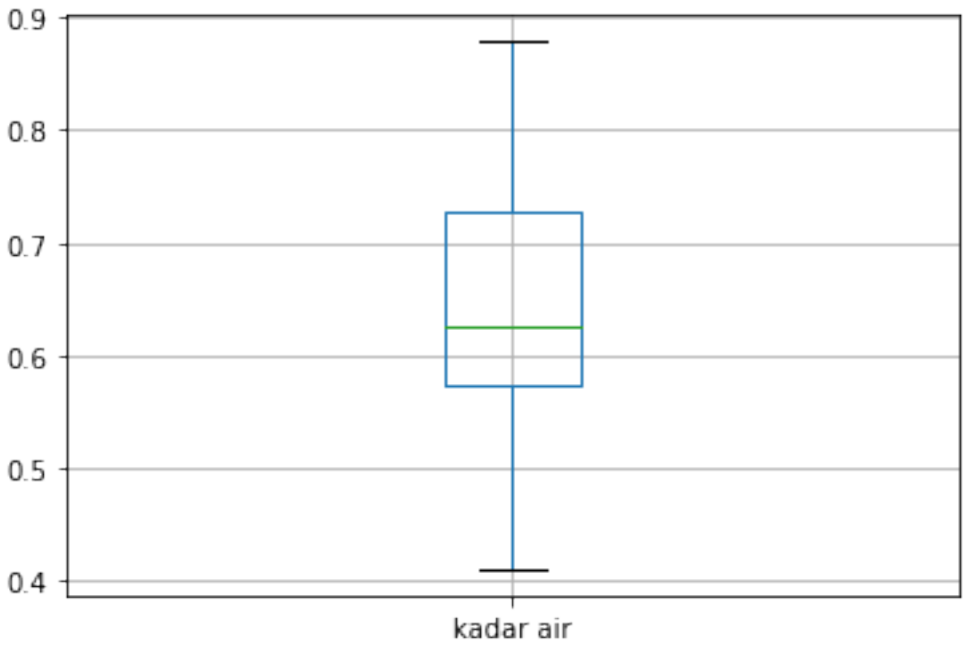
[24]:
```
df = pd.DataFrame(data = gandum, columns = ['area bulatan'])
boxplot = df.boxplot(column = ['area bulatan'])
```



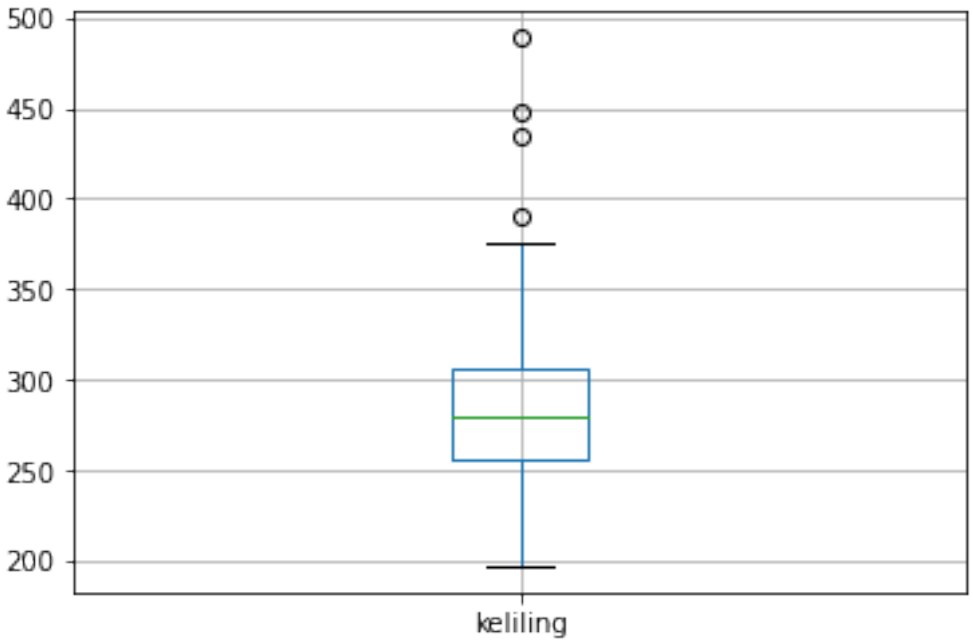
[25]:
```
df = pd.DataFrame(data = gandum, columns = ['diameter'])
boxplot = df.boxplot(column = ['diameter'])
```



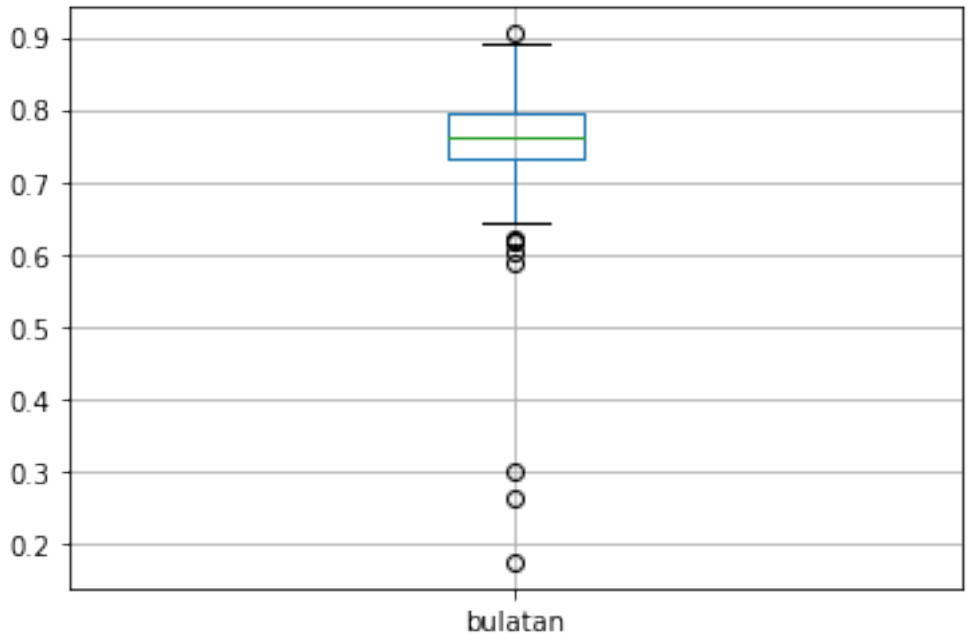
[26]:
```
df = pd.DataFrame(data = gandum, columns = ['kadar air'])
boxplot = df.boxplot(column = ['kadar air'])
```



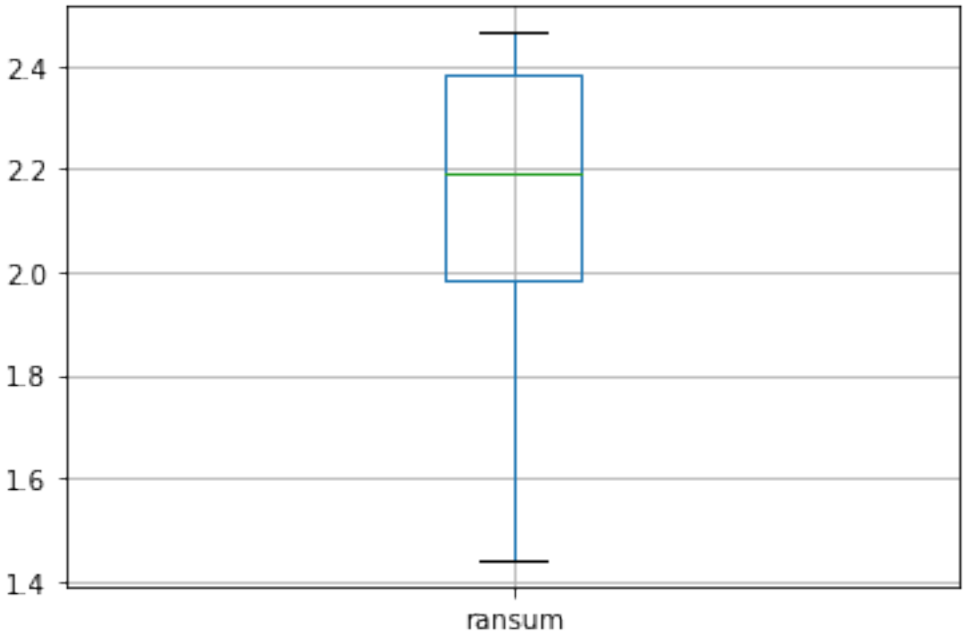
[27]:
```
df = pd.DataFrame(data = gandum, columns = ['keliling'])
boxplot = df.boxplot(column = ['keliling'])
```



[28]:
```
df = pd.DataFrame(data = gandum, columns = ['bulatan'])
boxplot = df.boxplot(column = ['bulatan'])
```



[29]:
```
df = pd.DataFrame(data = gandum, columns = ['ransum'])
boxplot = df.boxplot(column = ['ransum'])
```

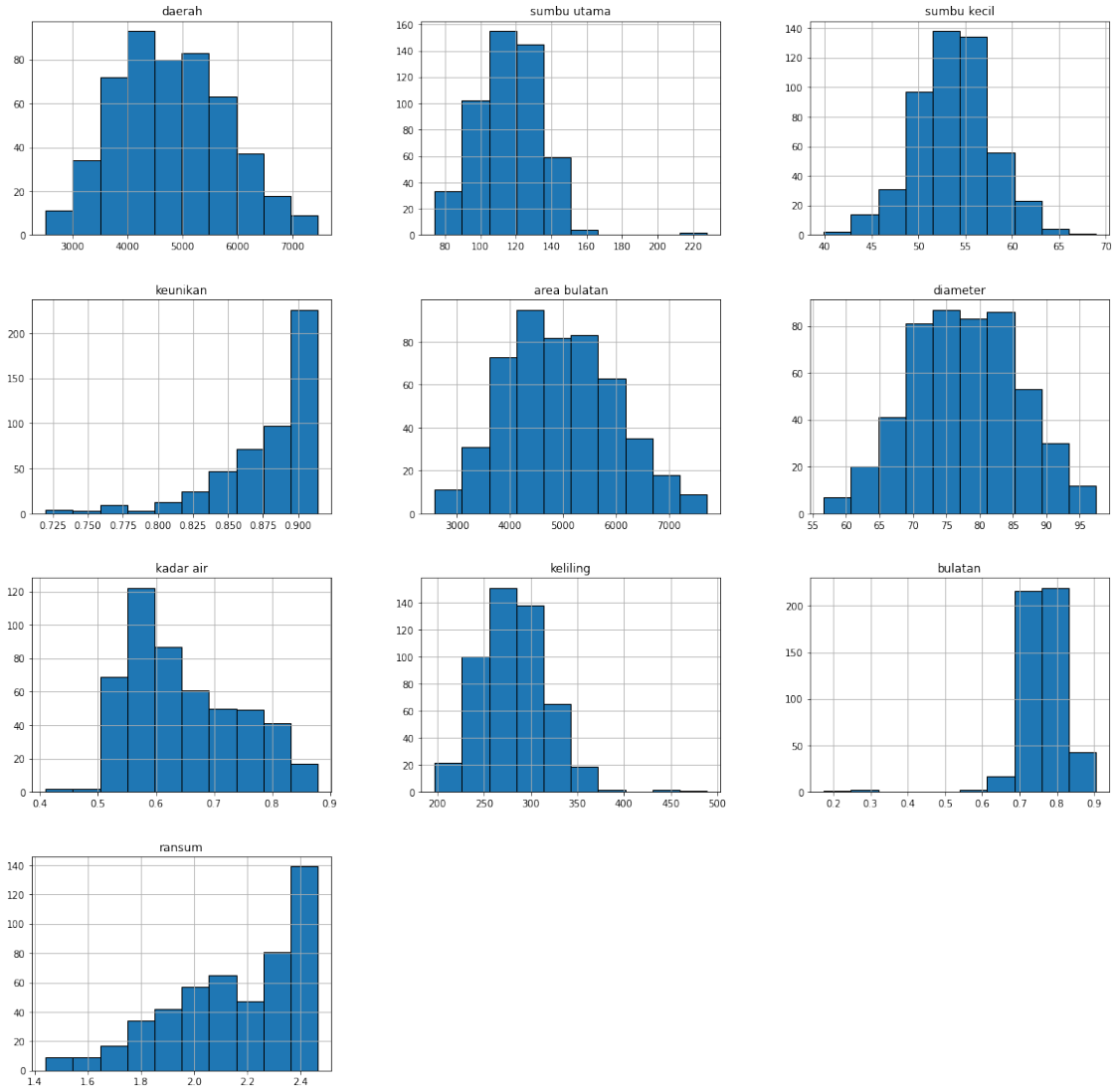


Histogram

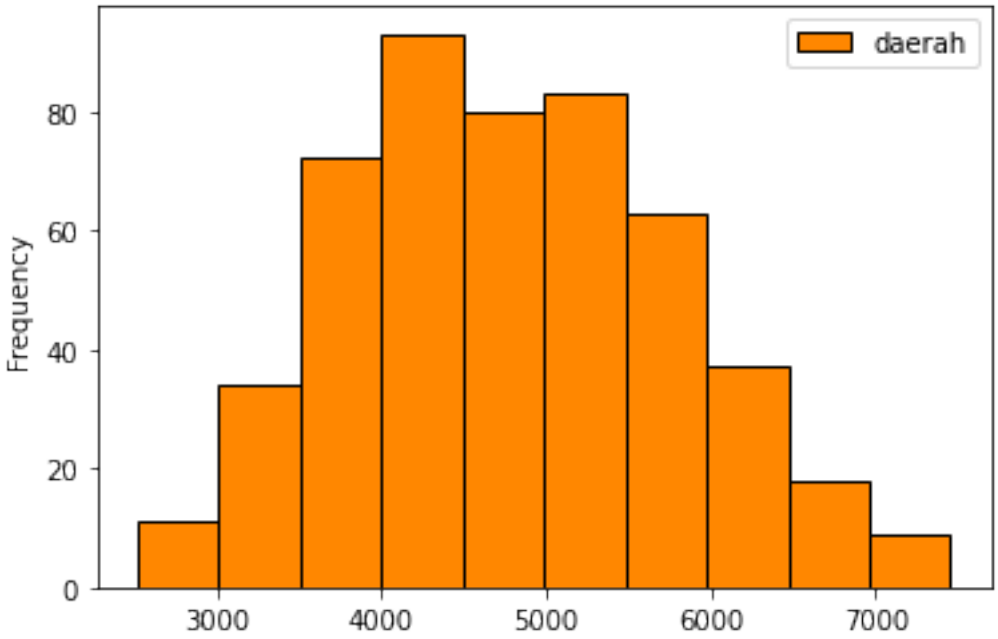
[30]:
```
gandum.hist(figsize=(20, 20), edgecolor='black')
```

```
C:\Users\Girvin Junod\AppData\Local\Programs\Python\Python39\lib\site-
packages\pandas\plotting\_matplotlib\tools.py:400: MatplotlibDeprecationWarning:
The is_first_col function was deprecated in Matplotlib 3.4 and will be removed
two minor releases later. Use ax.get_subplotspec().is_first_col() instead.
  if ax.is_first_col():
```

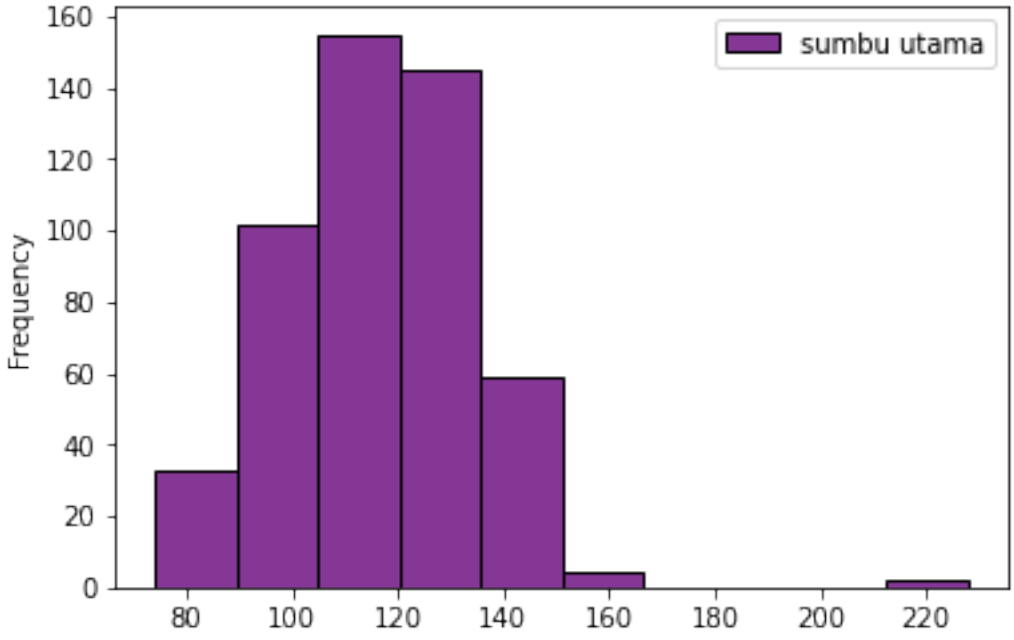
[30]:
```
array([[<AxesSubplot:title={'center':'daerah'}>,
        <AxesSubplot:title={'center':'sumbu utama'}>,
        <AxesSubplot:title={'center':'sumbu kecil'}>],
       [<AxesSubplot:title={'center':'keunikan'}>,
        <AxesSubplot:title={'center':'area bulatan'}>,
        <AxesSubplot:title={'center':'diameter'}>],
       [<AxesSubplot:title={'center':'kadar air'}>,
        <AxesSubplot:title={'center':'keliling'}>,
        <AxesSubplot:title={'center':'bulatan'}>],
       [<AxesSubplot:title={'center':'ransum'}>, <AxesSubplot:>,
        <AxesSubplot:>]], dtype=object)
```



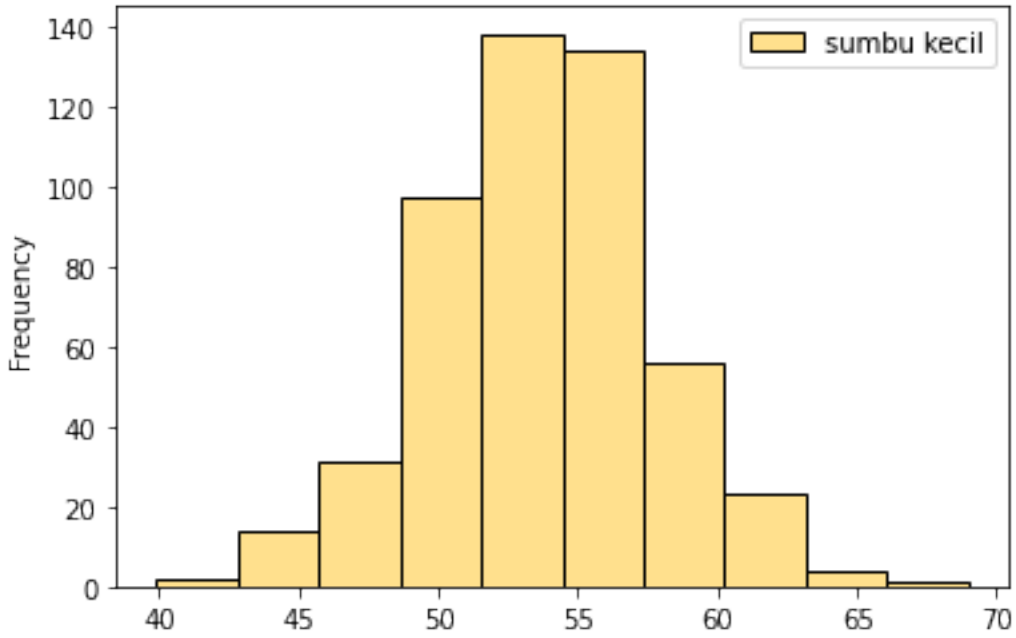
[31]:
```
df = pd.DataFrame(data = gandum, columns = ['daerah'])
hist = df.plot.hist(bins = 10, color = '#FF8700', edgecolor='black')
```



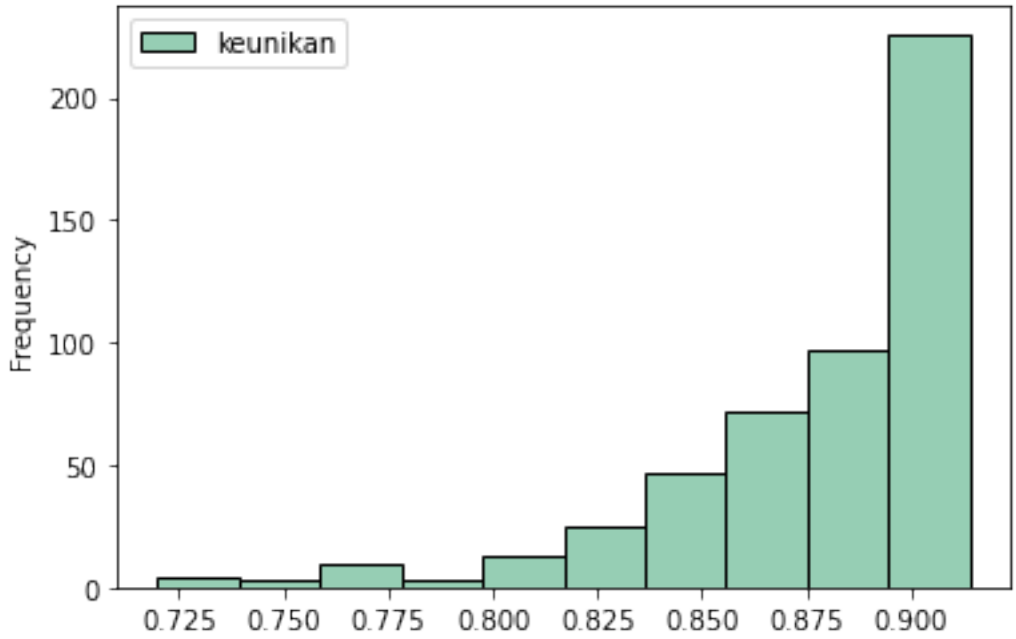
[32]:
```
df = pd.DataFrame(data = gandum, columns = ['sumbu utama'])
hist = df.plot.hist(bins = 10, color = '#843795', edgecolor='black')
```



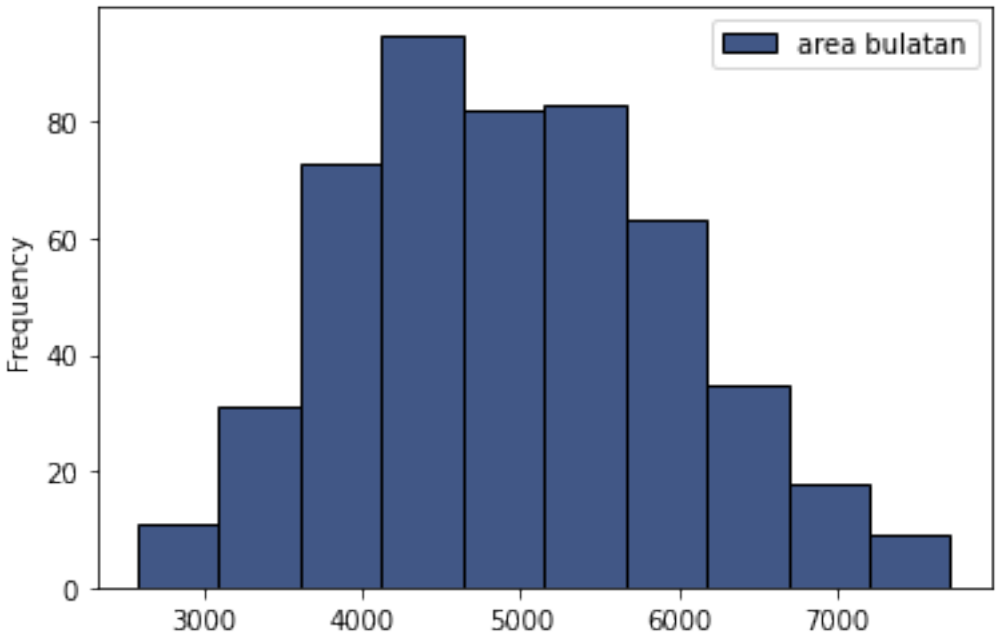
[33]:
```
df = pd.DataFrame(data = gandum, columns = ['sumbu kecil'])
hist = df.plot.hist(bins = 10, color = '#ffe08d', edgecolor='black')
```



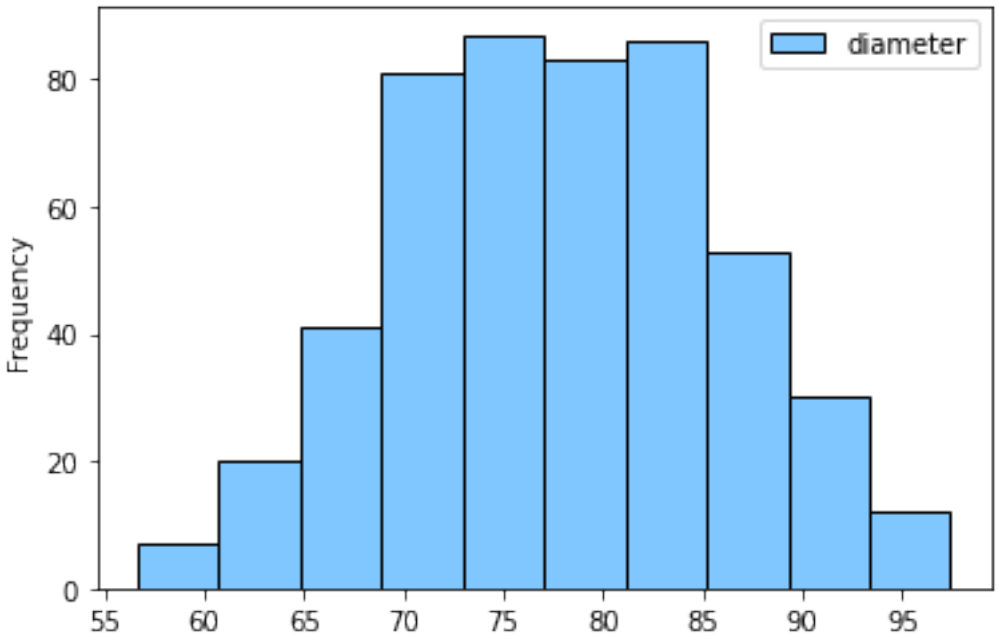
[34]:
```
df = pd.DataFrame(data = gandum, columns = ['keunikan'])
hist = df.plot.hist(bins = 10, color = '#96ceb4', edgecolor='black')
```



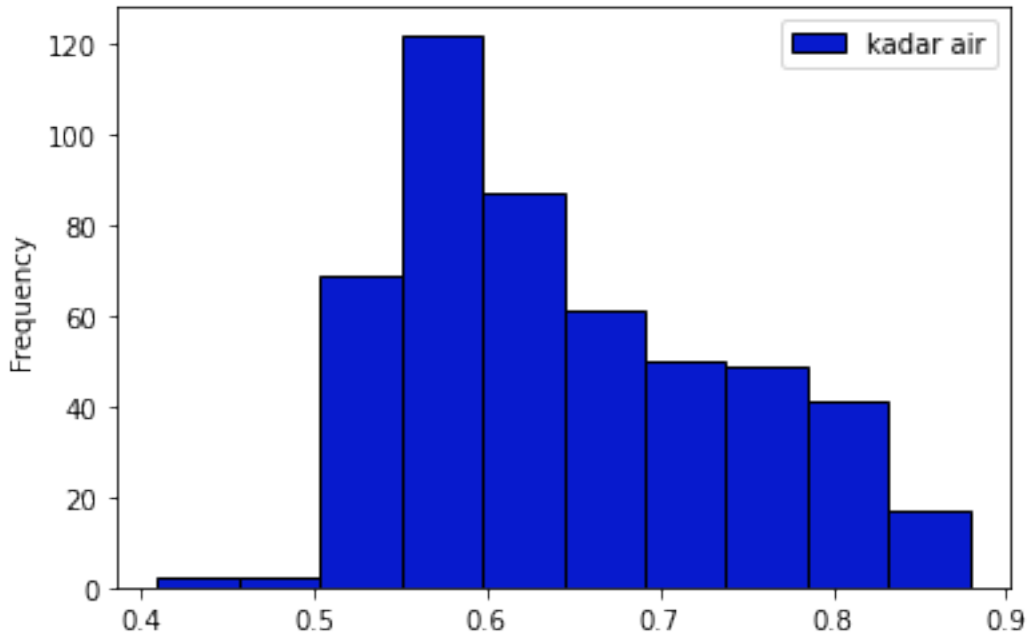
[35]:
```
df = pd.DataFrame(data = gandum, columns = ['area bulatan'])
hist = df.plot.hist(bins = 10, color = '#415786', edgecolor='black')
```



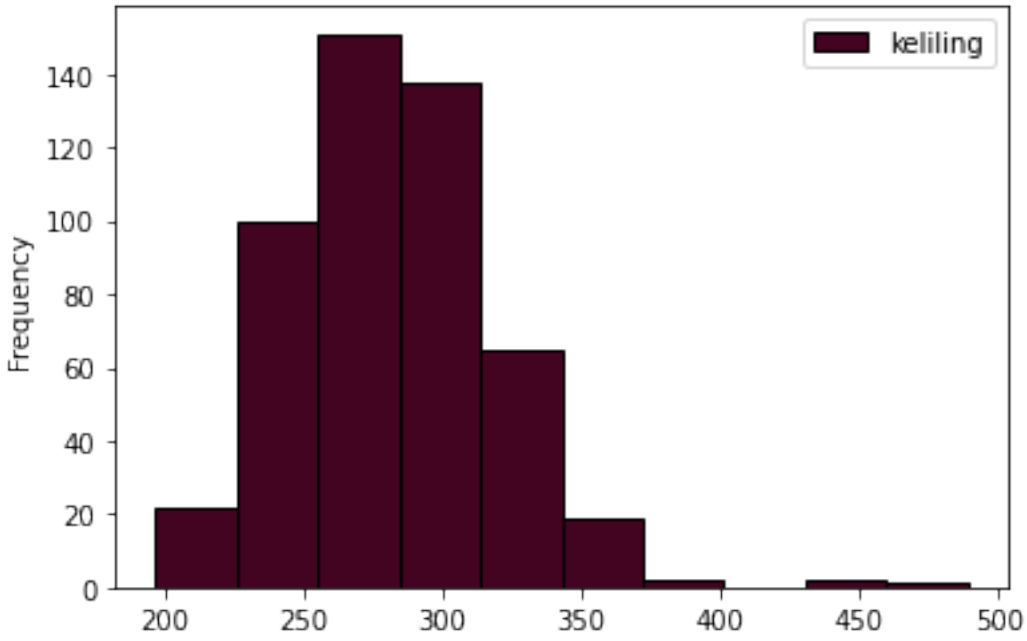
[36]:
```
df = pd.DataFrame(data = gandum, columns = ['diameter'])
hist = df.plot.hist(bins = 10, color = '#80C6FF', edgecolor='black')
```



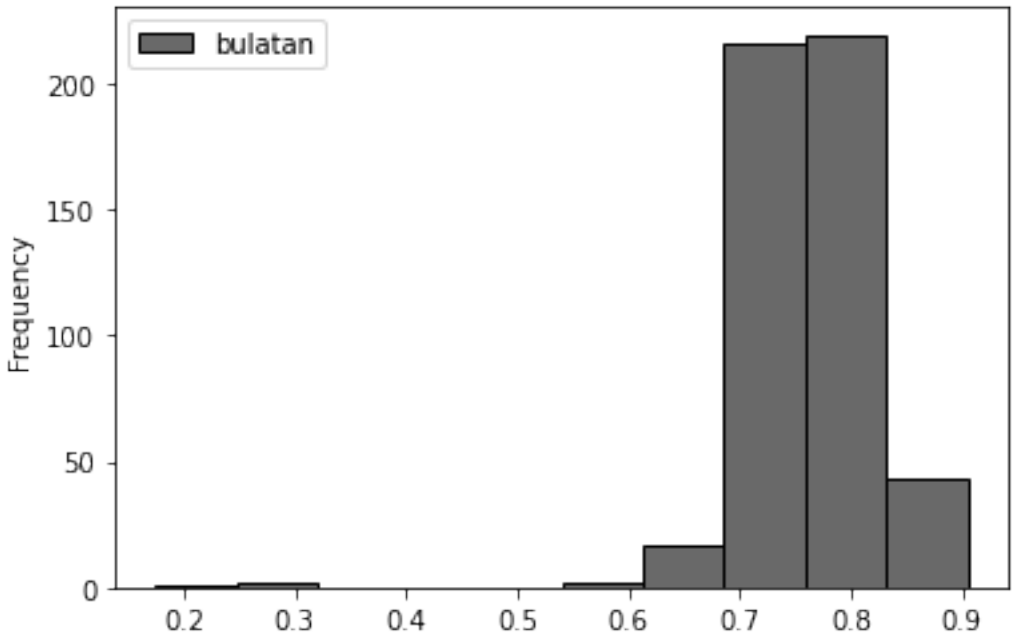
[37]:
```
df = pd.DataFrame(data = gandum, columns = ['kadar air'])
hist = df.plot.hist(bins = 10, color = '#071ACD', edgecolor='black')
```



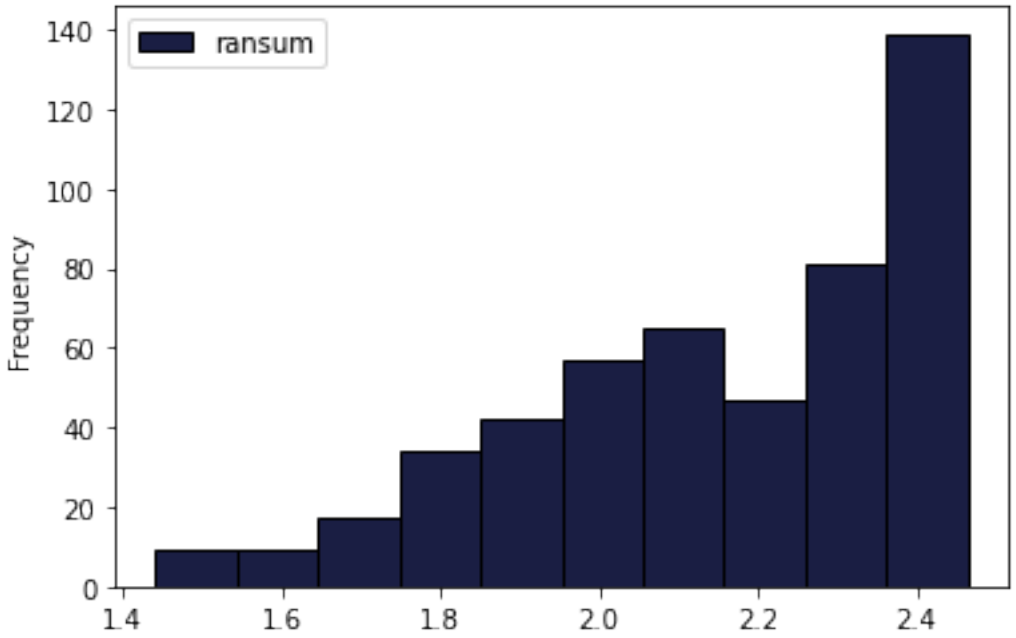
[38]:
```
df = pd.DataFrame(data = gandum, columns = ['keliling'])
hist = df.plot.hist(bins = 10, color = '#420420', edgecolor='black')
```



[39]:
```
df = pd.DataFrame(data = gandum, columns = ['bulatan'])
hist = df.plot.hist(bins = 10, color = '#696969', edgecolor='black')
```



[40]:
```
df = pd.DataFrame(data = gandum, columns = ['ransum'])
hist = df.plot.hist(bins = 10, color = '#1A1E43', edgecolor='black')
```



1.2 3. Normality Test

1.2.1 Daerah

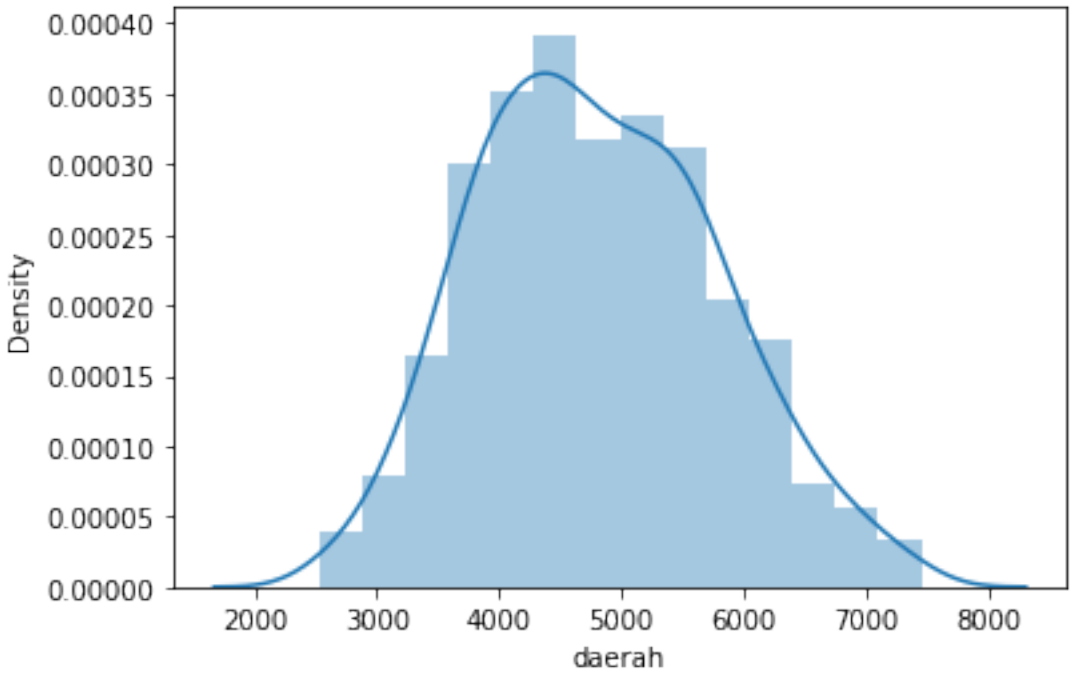
[41]:
```
df = pd.DataFrame(data = gandum, columns = ['daerah'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 0.0032707
Tidak berdistribusi normal
```

[42]:
```
sns.distplot(gandum['daerah'])
```

```
C:\Users\Girvin Junod\AppData\Local\Programs\Python\Python39\lib\site-
packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

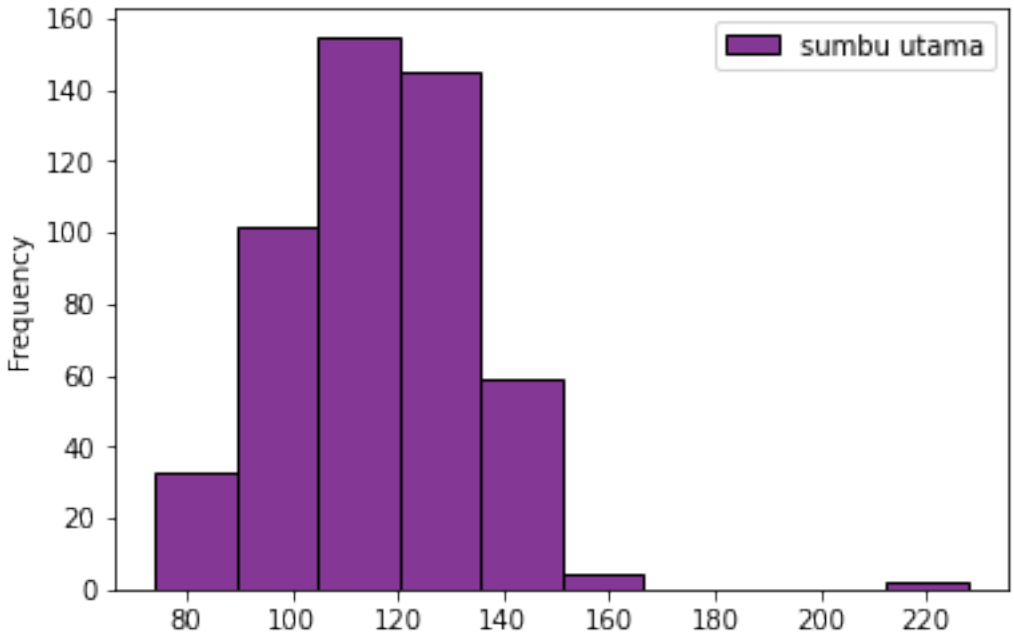
[42]: <AxesSubplot:xlabel='daerah', ylabel='Density'>



Penjelasan :

1.2.2 Sumbu Utama

[43]:
```
df = pd.DataFrame(data = gandum, columns = ['sumbu utama'])
hist = df.plot.hist(bins = 10, color = '#843795', edgecolor='black')
```

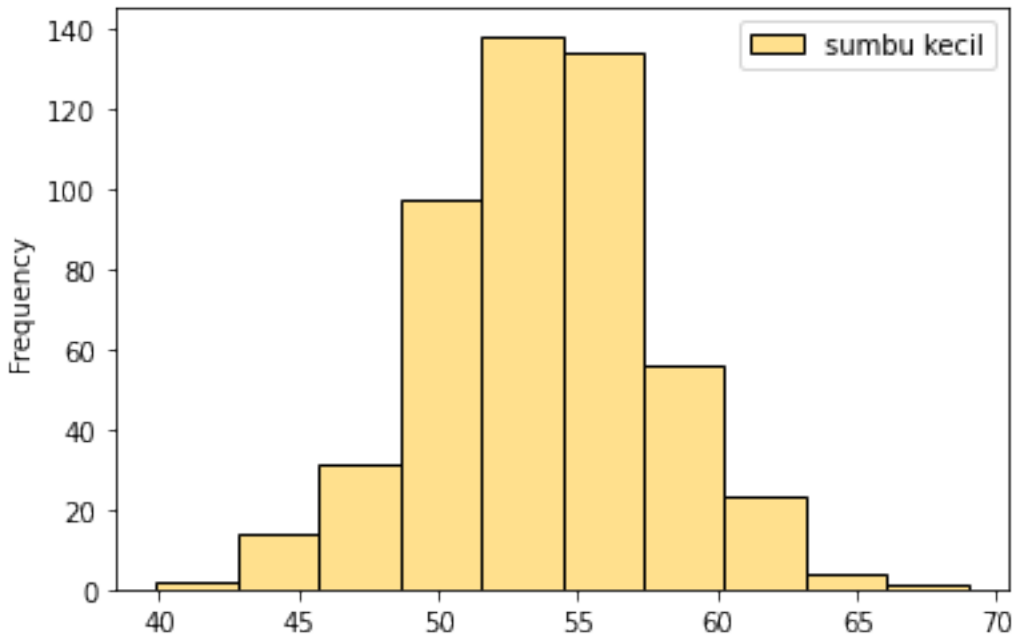


[44]:
```
df = pd.DataFrame(data = gandum, columns = ['sumbu utama'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 9.2362e-12
Tidak berdistribusi normal
```

1.2.3 Sumbu Kecil

[45]:
```
df = pd.DataFrame(data = gandum, columns = ['sumbu kecil'])
hist = df.plot.hist(bins = 10, color = '#ffe08d', edgecolor='black')
```

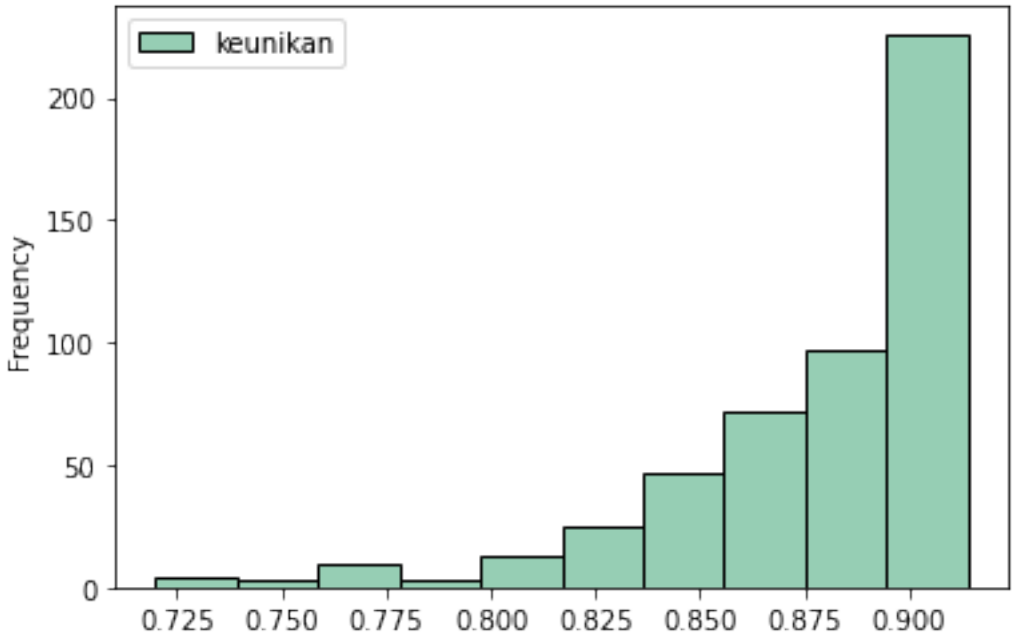


[46]:
```
df = pd.DataFrame(data = gandum, columns = ['sumbu kecil'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 0.423456
Berdistribusi normal
```

1.2.4 Keunikan

[47]:
```
df = pd.DataFrame(data = gandum, columns = ['keunikan'])
hist = df.plot.hist(bins = 10, color = '#96ceb4', edgecolor='black')
```

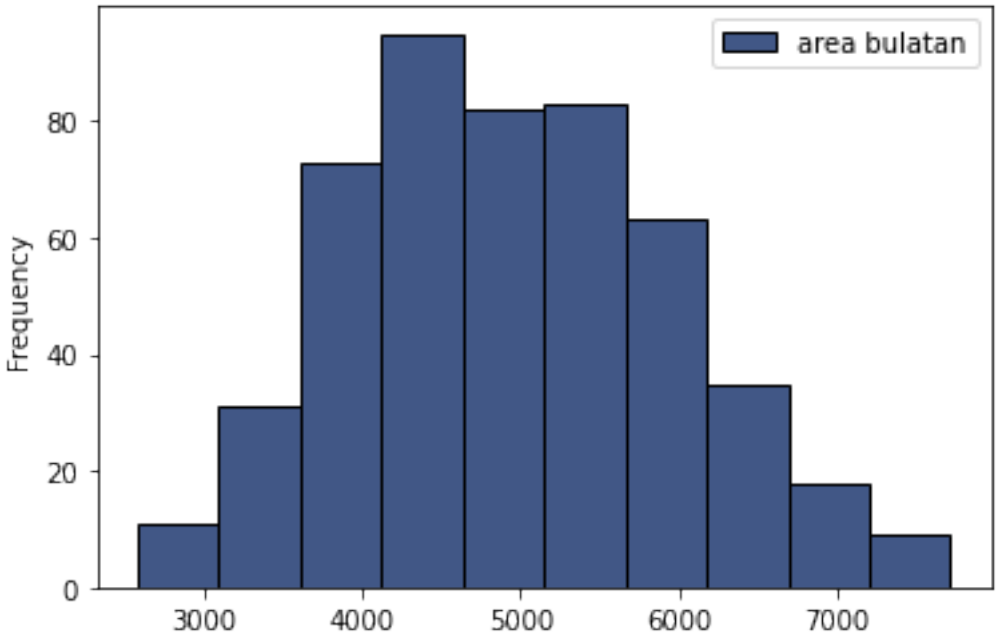


[48]:
```
df = pd.DataFrame(data = gandum, columns = ['keunikan'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 1.31517e-22
Tidak berdistribusi normal
```

1.2.5 AreaBulatan

[49]:
```
df = pd.DataFrame(data = gandum, columns = ['area bulatan'])
hist = df.plot.hist(bins = 10, color = '#415786', edgecolor='black')
```

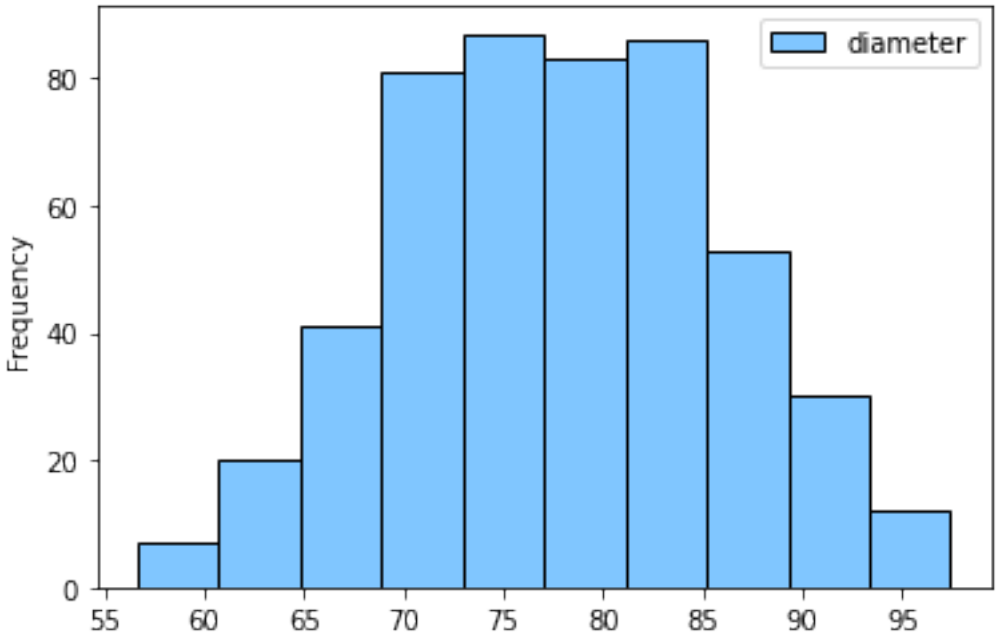


[50]:
```
df = pd.DataFrame(data = gandum, columns = ['area bulatan'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 0.00248471
Tidak berdistribusi normal
```

1.2.6 Diameter

[51]:
```
df = pd.DataFrame(data = gandum, columns = ['diameter'])
hist = df.plot.hist(bins = 10, color = '#80C6FF', edgecolor='black')
```

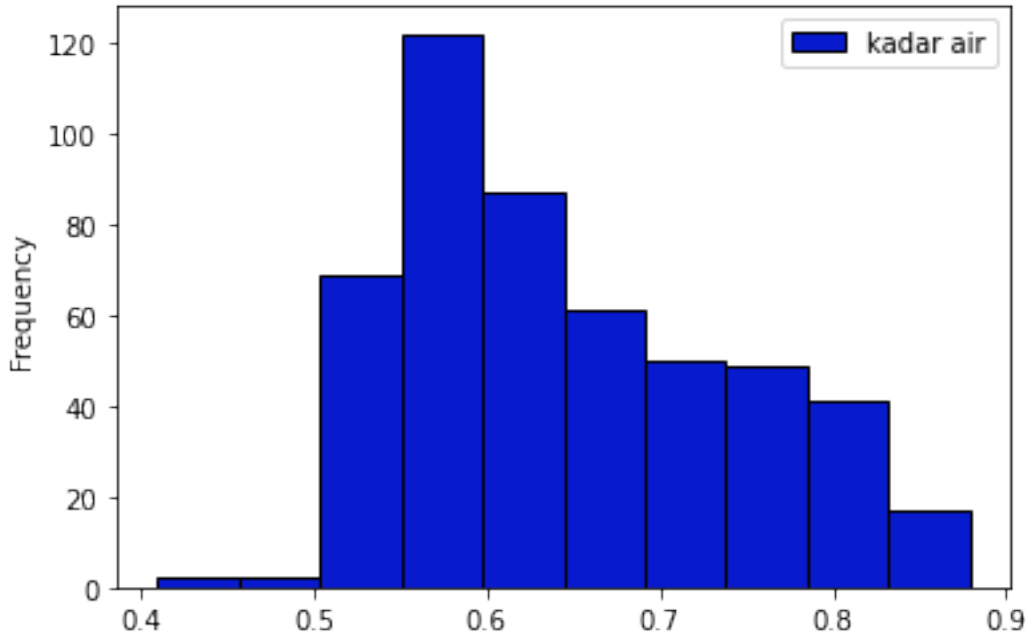


[52]:
```
df = pd.DataFrame(data = gandum, columns = ['diameter'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 0.118345
Berdistribusi normal
```

1.2.7 KadarAir

[53]:
```
df = pd.DataFrame(data = gandum, columns = ['kadar air'])
hist = df.plot.hist(bins = 10, color = '#071ACD', edgecolor='black')
```

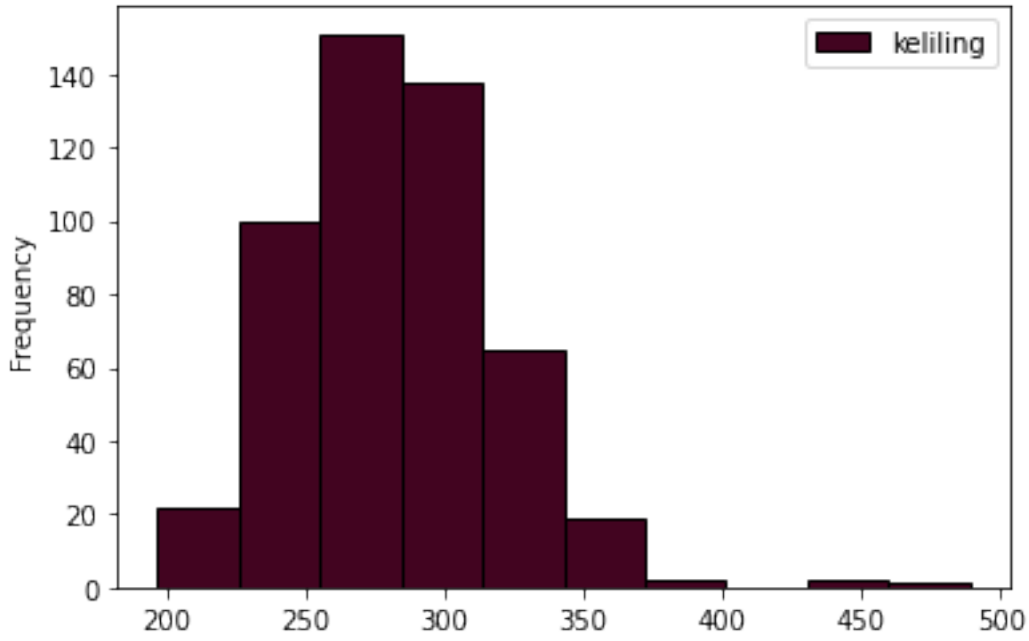


[54]:
```
df = pd.DataFrame(data = gandum, columns = ['kadar air'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 1.9595e-12
Tidak berdistribusi normal
```

###Keliling

[55]:
```
df = pd.DataFrame(data = gandum, columns = ['keliling'])
hist = df.plot.hist(bins = 10, color = '#420420', edgecolor='black')
```

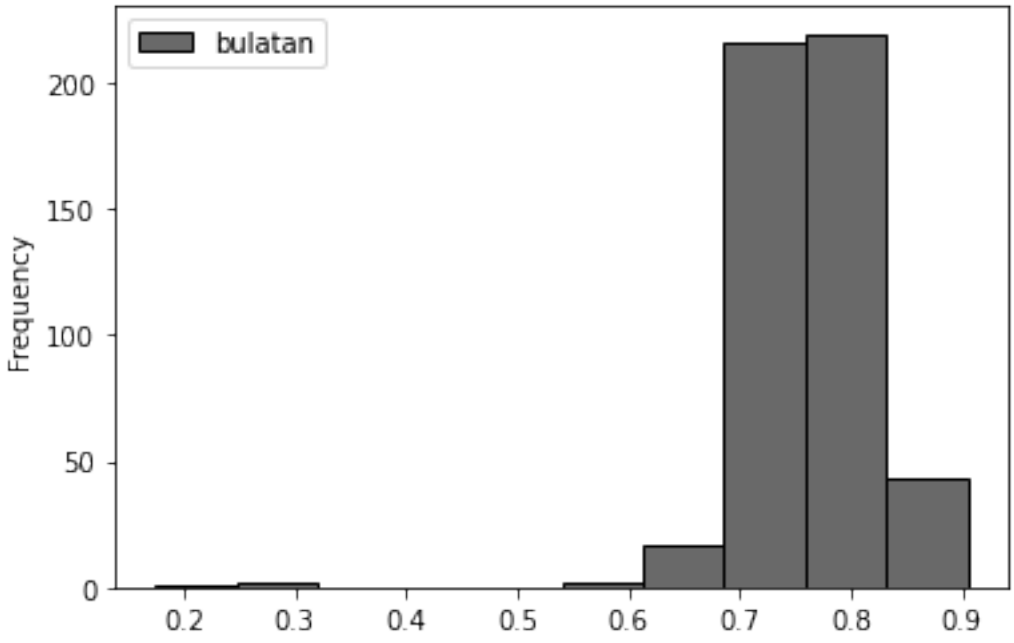


[56]:
```
df = pd.DataFrame(data = gandum, columns = ['keliling'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 9.72839e-09
Tidak berdistribusi normal
```

1.2.8 Bulatan

[57]:
```
df = pd.DataFrame(data = gandum, columns = ['bulatan'])
hist = df.plot.hist(bins = 10, color = '#696969', edgecolor='black')
```

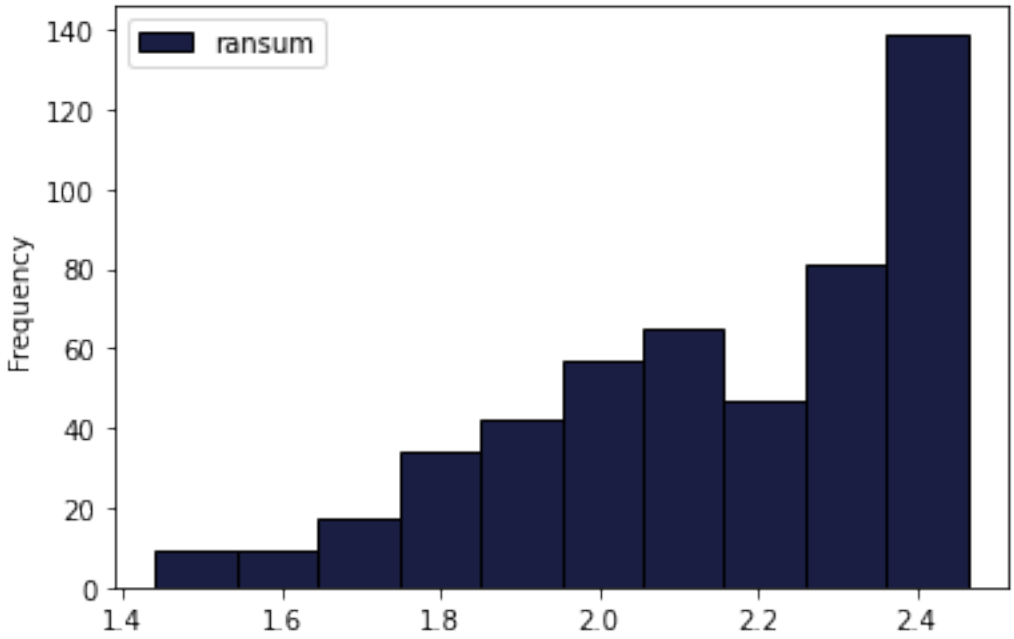


[58]:
```
df = pd.DataFrame(data = gandum, columns = ['bulatan'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 6.89916e-26
Tidak berdistribusi normal
```

1.2.9 Ransum

[59]:
```
df = pd.DataFrame(data = gandum, columns = ['ransum'])
hist = df.plot.hist(bins = 10, color = '#1A1E43', edgecolor='black')
```



[60]:
```
df = pd.DataFrame(data = gandum, columns = ['ransum'])
#stat, p = s.normaltest(df, axis= 0, nan_policy='omit') D'Agostino
stat, p = s.shapiro(df) #shapiro wilks
print("p = {:g}".format(p))
if p <= alpha:  # null hypothesis: x comes from a normal distribution
    print("Tidak berdistribusi normal")
else:
    print("Berdistribusi normal")
```

```
p = 6.24554e-15
Tidak berdistribusi normal
```

2 Test Hipotesis 1 Sampel

[61]:
```
import math

def zscore(x, mean, std, n):
  return (x-mean)*math.sqrt(n)/std
```

[62]:
```
def ptoz(p):
  return s.norm.ppf(p)
def ztop(z):
  return s.norm.cdf(z)
```

##(**A**) Nilai rata-rata Daerah di atas 4700?

Langkah-Langkah

1. Tentukan hipotesis nol

   H0 : miu = 4700

2. Tentukan hipotesis alternatif

   H1 : miu > 4700

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji statistik yang digunakan adalah : Uji parameter populasi pengujian rataan satu sampel one tail test variance known

   Daerah kritis : z > zalpha : z > 1.645

5. Hitung nilai uji statistik

[63]:
```
mean = 4700
rataan = gandum['daerah'].mean()
std = gandum['daerah'].std()
n = gandum['daerah'].count()
z = zscore(rataan, mean, std, n)
p = 1 - ztop(z)
zalpha = ptoz(1-alpha)
```

2.0.1 6. Ambil keputusan

[64]:
```
if z > zalpha:
  print("Tolak null hypothesis")
  print("Hipotesis alternatif benar")
  print("Nilai rata-rata Daerah di atas 4700")
else:
  print("Terima null hypothesis")
  print("Hipotesis alternatif salah")
```

```
Tolak null hypothesis
Hipotesis alternatif benar
Nilai rata-rata Daerah di atas 4700
```

2.1 B) Nilai Rata-rata Sumbu Utama tidak sama dengan 116?

Langkah-Langkah

1. Tentukan hipotesis nol

   H0 : miu = 116

2. Tentukan hipotesis alternatif

H1 : miu != 116 (two-tailed test)

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji statistik : Uji parameter populasi pengujian rataan satu sampel two tail test variance known

   daerah kritis : z < (-zalpha/2) or z > (zalpha/2)

5. Hitung nilai uji statistik

[65]:
```
mean = 116
rataan = gandum['sumbu utama'].mean()
std = gandum['sumbu utama'].std()
n = gandum['sumbu utama'].count()
z = zscore(rataan, mean, std, n)
zalpha = ptoz(alpha/2)
```

6. Ambil Keputusan

[66]:
```
if z < zalpha or z > -zalpha:
  print("Null hypothesis ditolak")
else:
  print("Null hypothesis diterima")
  print("Rata-rata sumbu utama sama dengan 116")
```

```
Null hypothesis diterima
Rata-rata sumbu utama sama dengan 116
```

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

Langkah-Langkah

1. Tentukan hipotesis nol

   H0 : miu = 50

2. Tentukan hipotesis alternatif

   H1 : miu != 50 (two-tailed test)

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji statistik : Uji parameter populasi pengujian rataan satu sampel two tail test variance known

   daerah kritis : z < (-zalpha/2) or z > (zalpha/2)

5. Hitung nilai uji statistik

[67]:
```
mean = 50
sampel = gandum['sumbu utama'].head(20)
rataan = sampel.mean()
std = sampel.std()
n = sampel.count()
z = zscore(rataan, mean, std, n)
zalpha = ptoz(alpha/2)
```

6. Ambil Keputusan

[68]:
```
if z < zalpha or z > -zalpha:
  print("Null hypothesis ditolak")
  print("Nilai rata-rata 20 baris pertama kolom sumbu kecil bukan 50")
else:
  print("Null hypothesis diterima")
```

```
Null hypothesis ditolak
Nilai rata-rata 20 baris pertama kolom sumbu kecil bukan 50
```

2.3 D) Proporsi nilai Diameter yang lebih dari 85, adalah tidak sama dengan 15% ?

Langkah-Langkah

1. Tentukan hipotesis nol

   H0 : p = 0.15

2. Tentukan hipotesis alternatif

   H1 : p != 0.15 (two-tailed test)

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji satu parameter populasi pengujian proporsi satu sampel dengan sample banyak

   Daerah kritis : z > zalpha/2 or z < -zalpha/2

5. Hitung nilai statistik

[69]:
```
diameter = gandum['diameter'].loc[gandum['diameter'] > 85]
p0 = 0.15
q0 = 1 - p0
zalpha = ptoz(alpha/2)
n = gandum['diameter'].count()
x = diameter.count()
p1 = x/n

z = (p1-p0) * math.sqrt(n)/ math.sqrt(p0*q0)
```

6. Ambil Keputusan

[70]:
```
if z < zalpha or z > -zalpha:
  print("Null hypothesis ditolak")
  print("Nilai Diameter yang lebih dari 85, adalah tidak sama dengan 15%")
else:
  print("Null hypothesis diterima")
```

```
Null hypothesis ditolak
Nilai Diameter yang lebih dari 85, adalah tidak sama dengan 15%
```

2.4 E) Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari 5% ?

Langkah-Langkah

1. Tentukan hipotesis nol

   H0 : p = 0.05

2. Tentukan hipotesis alternatif

   H1 : p < 0.05 (one-tailed test)

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji satu parameter populasi pengujian proporsi satu sampel dengan sample banyak one tailed test

   Daerah kritis : z < -zalpha

5. Hitung nilai uji statistik

[71]:
```
keliling = gandum['keliling'].loc[gandum['keliling'] < 100]
p0 = 0.05
q0 = 1 - p0
zalpha = ptoz(alpha)
n = gandum['keliling'].count()
x = keliling.count()
p1 = x/n

z = (p1-p0) * math.sqrt(n)/ math.sqrt(p0*q0)
```

6. Ambil Keputusan

[72]:
```
if z < -zalpha:
  print("Null hypothesis ditolak")
  print("Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari 5%")
else:
  print("Null hypothesis diterima")
```

```
Null hypothesis ditolak
Proporsi nilai Keliling yang kurang dari 100, adalah kurang dari 5%
```

3 Test Hipotesis 2 Sampel

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

[73]:
```
def ztwotestmean(d0, x1, x2, var1, var2, n1, n2):
  return((x1 - x2) - d0)/math.sqrt((var1/n1) + (var2/n2))
```

Langkah-Langkah

1. Tentukan hipotesis nol

   H0 : miu1 - miu2 = 0

2. Tentukan hipotesis alternatif

   H1 : miu1 - miu2 != 0 (two-tailed)

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji statistik : two sample two tailed mean test known std

   daerah kritis : z < -zalpha/2 or z > zalpha/2

5. Hitung nilai uji statistik

[74]:
```
areaBulat = gandum['area bulatan']

d0 = 0

zalpha = ptoz(alpha/2)
bagian1 = areaBulat.head(areaBulat.size // 2)
bagian2 = areaBulat.tail(areaBulat.size // 2)

mean1 = bagian1.mean()
mean2 = bagian2.mean()

var1 = bagian1.var()
var2 = bagian2.var()

z = ztwotestmean(d0, mean1, mean2, var1, var2, bagian1.size, bagian2.size)
```

6. Ambil Keputusan

[75]:
```
if z < zalpha or z > -zalpha:
  print("Null hypothesis ditolak")
  print("Rata-rata kedua bagian tidak sama")
else:
  print("Null hypothesis diterima")
```

```
Null hypothesis ditolak
Rata-rata kedua bagian tidak sama
```

3.2 B) 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-Langkah

1. Tentukan hipotesis nol

   H0 : miu1 - miu2 = 0.2

2. Tentukan hipotesis alternatif

   H1 : miu1 - miu2 > 0.2 (one-tailed)

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji statistik : two sample two tailed mean test known std

   Daerah kritis : z > zalpha

5. Hitung nilai uji statistik

[76]:
```
kadarAir = gandum['kadar air']

d0 = 0.2

zalpha = ptoz(alpha)
bagian1 = kadarAir.head(kadarAir.size // 2)
bagian2 = kadarAir.tail(kadarAir.size // 2)

mean1 = bagian1.mean()
mean2 = bagian2.mean()

var1 = bagian1.var()
var2 = bagian2.var()

z = ztwotestmean(d0, mean1, mean2, var1, var2, bagian1.size, bagian2.size)
```

6. Ambil Keputusan

[77]:
```
if z > zalpha:
  print("Null hypothesis ditolak")
else:
  print("Null hypothesis diterima")
  print("Rata-rata bagian awal tidak lebih besar dari pada bagian akhir sebesar␣
 ↪0.2")
```

```
Null hypothesis diterima
Rata-rata bagian awal tidak lebih besar dari pada bagian akhir sebesar 0.2
```

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

Langkah-Langkah

1. Tentukan hipotesis nol

   H0 : miu1 - miu2 = 0

2. Tentukan hipotesis alternatif

   H1 : miu1 - miu2 != 0 (two-tailed)

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji statistik :

   Daerah kritis :

5. Hitung nilai uji statistik

[78]:
```
bulatan = gandum['bulatan']

bagian1 = bulatan.head(20)
bagian2 = bulatan.tail(20)

mean1 = bagian1.mean()
mean2 = bagian2.mean()

var1 = bagian1.var()
var2 = bagian2.var()

d0 = 0

zalpha = ptoz(alpha/2)
z = ztwotestmean(d0, mean1, mean2, var1, var2, bagian1.size, bagian2.size)
```

6. Ambil keputusan

[79]:
```
if z < zalpha or z > -zalpha:
  print("Null hypothesis ditolak")
  print("Rata-rata 20 baris pertama kolom bulatan tidak sama dengan 20 baris␣
 ↪terakhirnya")
else:
  print("Null hypothesis diterima")
```

```
Null hypothesis ditolak
Rata-rata 20 baris pertama kolom bulatan tidak sama dengan 20 baris terakhirnya
```

3.4 D) Proporsi nilai bagian awal Ransum yang lebih dari 2, adalah lebih besar daripada, proporsi nilai yang sama di bagian akhir Ransum?

Langkah-Langkah

1. Tentukan hipotesis nol

   H0 : p1 - p2 - 0

2. Tentukan hipotesis alternatif

   H1 : p1 - p2 > 0 (one-tailed)

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji statistik satu parameter populasi pengujian proporsi dua sampel one-tailed tes normal, z

   Daerah kritis : z > zalpha

5. Hitung nilai uji statistik

[80]:
```
ransum = gandum['ransum']

n1 = ransum.head(ransum.size // 2)
n2 = ransum.tail(ransum.size // 2)


x1 = n1.loc[n1 > 2]
x2 = n2.loc[n2 > 2]

p1 = x1.size/n1.size
p2 = x2.size/n2.size

p = (x1.size + x2.size)/(n1.size + n2.size)
q = 1 - p

pembilang = p1 - p2
penyebut = (p*q/n1.size) + (p*q/n2.size)
z = pembilang/math.sqrt(penyebut)
```

6. Ambil Keputusan

[81]:
```
if z > zalpha:
  print("Null hypothesis ditolak")
  print("Proporsi nilai bagian awal Ransum yang lebih dari 2, adalah lebih␣
 ↪besar daripada, proporsi nilai yang sama di bagian akhir Ransum")
else:
  print("Null hypothesis diterima")
```

```
Null hypothesis ditolak
Proporsi nilai bagian awal Ransum yang lebih dari 2, adalah lebih besar
daripada, proporsi nilai yang sama di bagian akhir Ransum
```

3.5 E) Bagian awal kolom Diameter memiliki variansi yang sama dengan bagian akhirnya?

Langkah-Langkah

1. Tentukan hipotesis nol

   H0 : var1 - var2 = 0

2. Tentukan hipotesis alternatif

   H1 : var1 - var2 != 0 (two-tailed)

3. Tentukan tingkat signifkan

   alpha = 0.05

4. Tentukan uji statistik yang sesuai dan tentukan daerah kritis.

   Uji statistik : distribusi F

   Daerah kritis : f < (1-aplha/2)f(v1,v2) or f > falpha/2(v1,v2)

5. Hitung nilai uji statistik

[82]:
```
diameter = gandum['diameter']

awal = diameter.head(diameter.size // 2)
akhir = diameter.tail(diameter.size // 2)

awalvar = awal.var()
akhirvar = akhir.var()

if(awalvar > akhirvar):
  f = akhirvar/awalvar
else:
  f = awalvar/akhirvar

f1 = s.f.ppf(q=1-alpha/2, dfn=awal.size-1, dfd=akhir.size-1)
f2 = s.f.ppf(q=alpha/2, dfn=awal.size-1, dfd=akhir.size-1)
```

6. Ambil Keputusan

[83]:
```
if(f > f1 or f < f2):
  print("Null hypothesis ditolak")
else:
  print("Null hypothesis diterima")
  print("Bagian awal kolom Diameter tidak memiliki variansi yang sama dengan␣
 ↪bagian akhirnya")
```

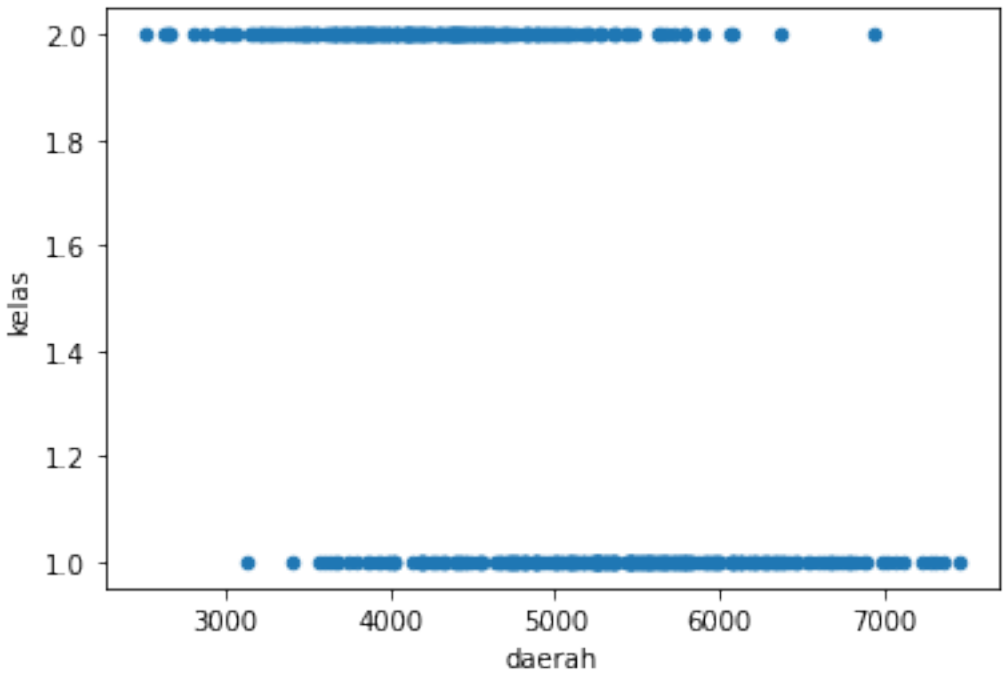
```
Null hypothesis diterima
Bagian awal kolom Diameter tidak memiliki variansi yang sama dengan bagian
akhirnya
```

4 Test Korelasi

4.0.1 Daerah dengan Kelas

[84]:
```
g.plot(kind='scatter', x='daerah', y='kelas')
a = g['daerah']
b = g['kelas']
a.corr(b)
```

[84]: -0.6027466517416662

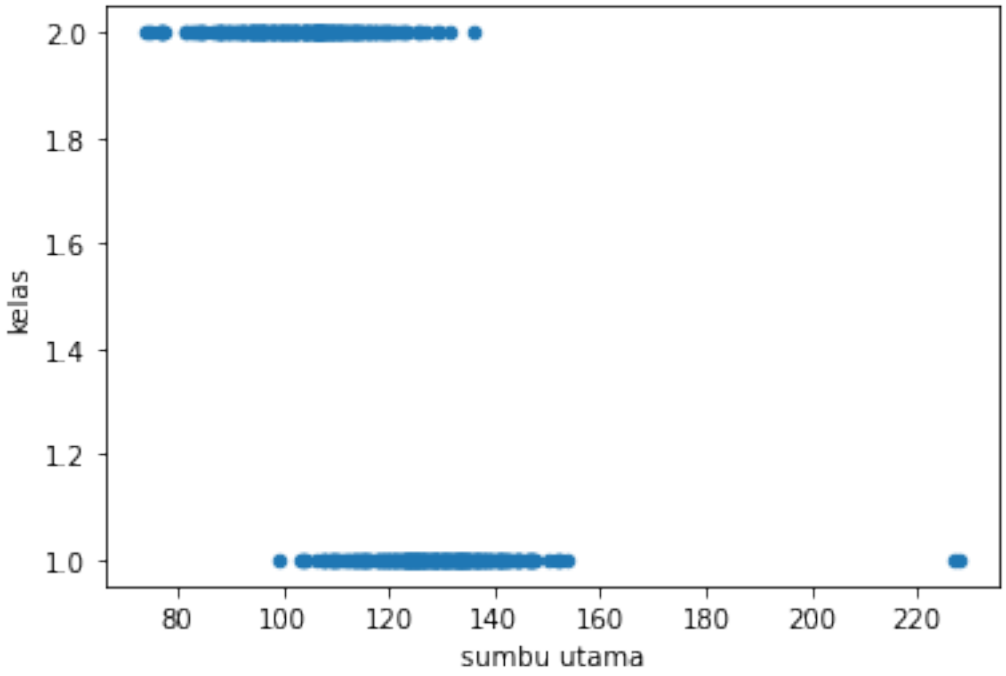


4.0.2 Kesimpulan :

4.0.3 Sumbu Utama dengan Kelas

[85]:
```
g.plot(kind='scatter', x='sumbu utama', y='kelas')
a = g['sumbu utama']
b = g['kelas']
a.corr(b)
```

[85]: -0.7130906104204593

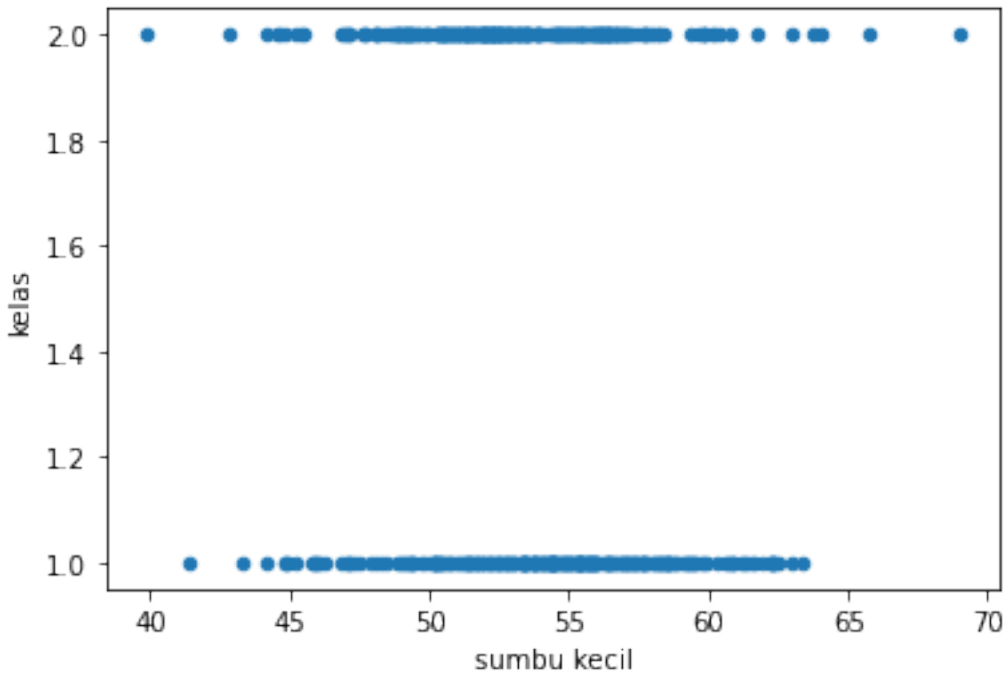


4.0.4 Kesimpulan :

4.0.5 Sumbu Kecil dengan Kelas

[86]:
```
g.plot(kind='scatter', x='sumbu kecil', y='kelas')
a = g['sumbu kecil']
b = g['kelas']
a.corr(b)
```

[86]: -0.15297517335535027

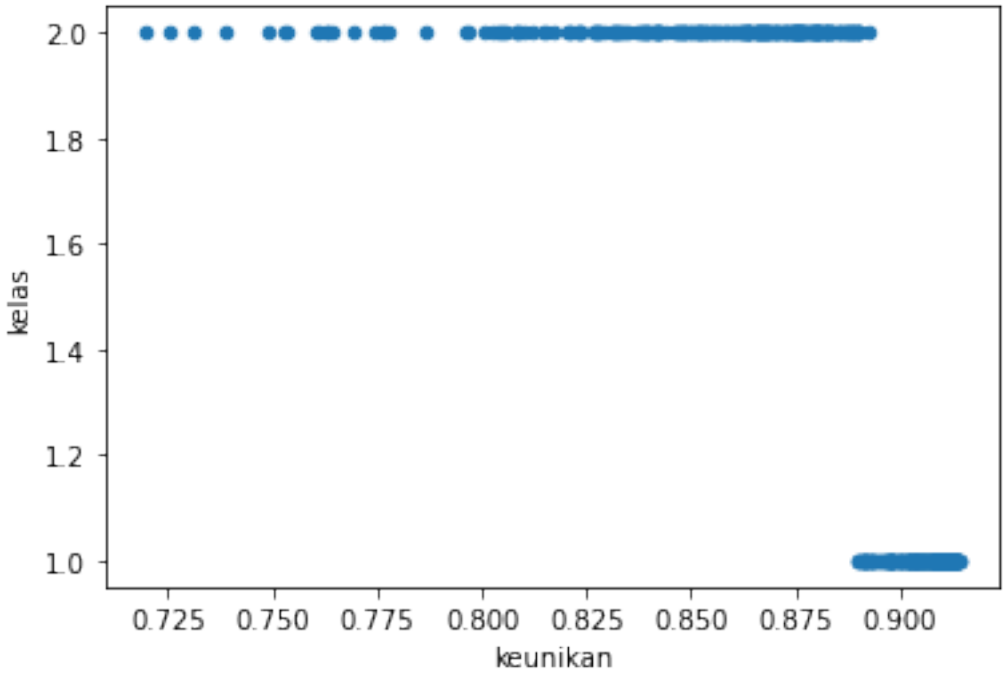


4.0.6 Kesimpulan :

4.0.7 Keunikan dengan Kelas

[87]:
```
g.plot(kind='scatter', x='keunikan', y='kelas')
a = g['keunikan']
b = g['kelas']
a.corr(b)
```

[87]: -0.7304563686511922

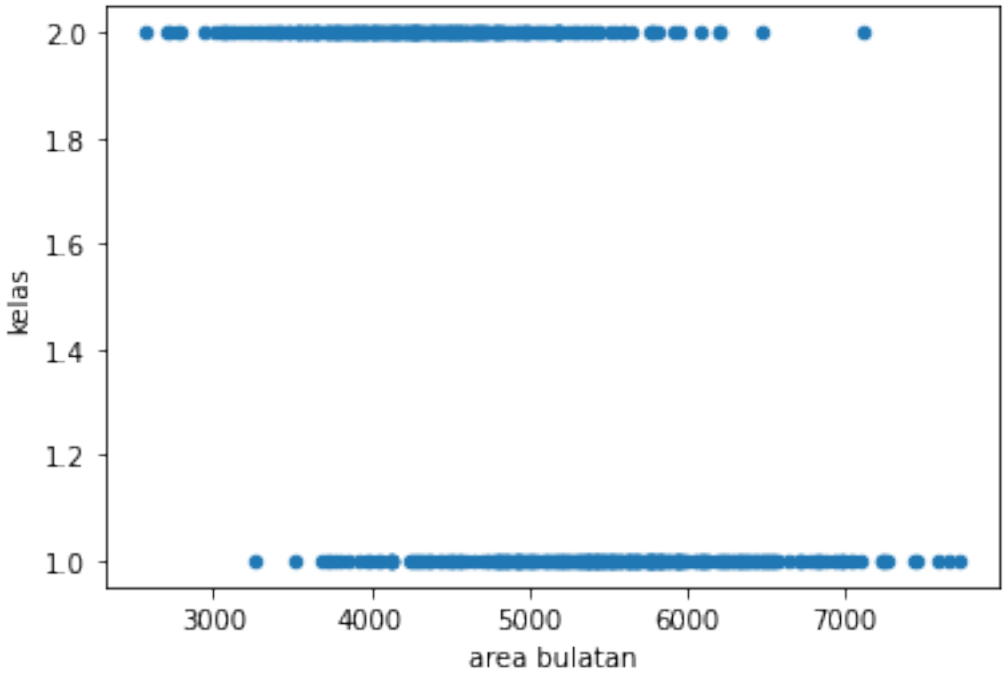


4.0.8 Kesimpulan :

4.0.9 Area Bulatan dengan Kelas

[88]:
```
g.plot(kind='scatter', x='area bulatan', y='kelas')
a = g['area bulatan']
b = g['kelas']
a.corr(b)
```

[88]: -0.6073125434153751

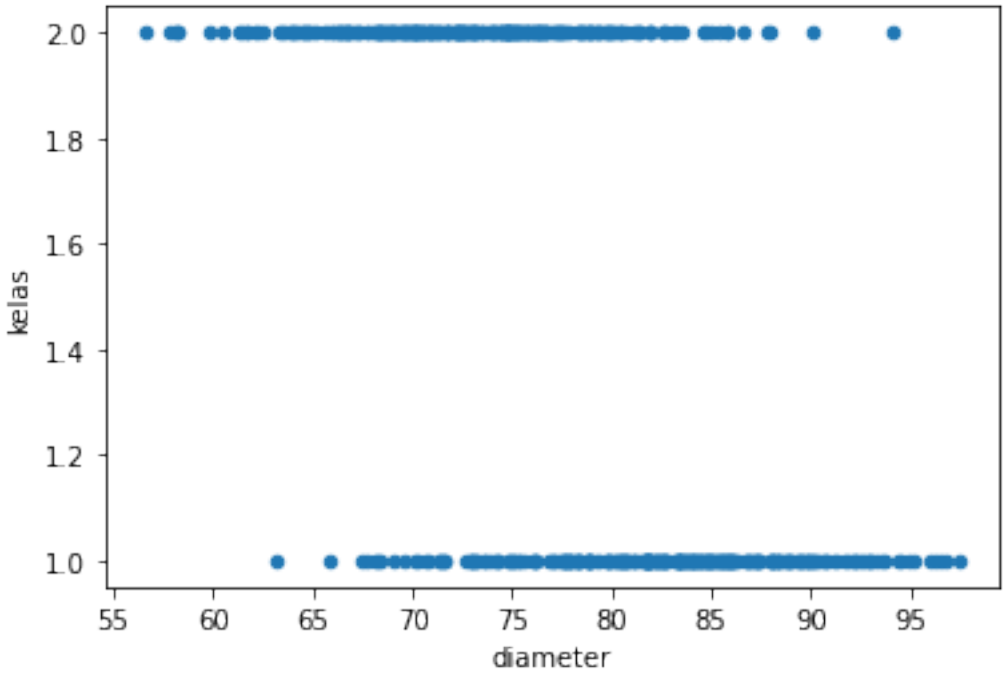


4.0.10 Kesimpulan :

4.0.11 Diameter dengan Kelas

[89]:
```
g.plot(kind='scatter', x='diameter', y='kelas')
a = g['diameter']
b = g['kelas']
a.corr(b)
```

[89]: -0.6025356896618813

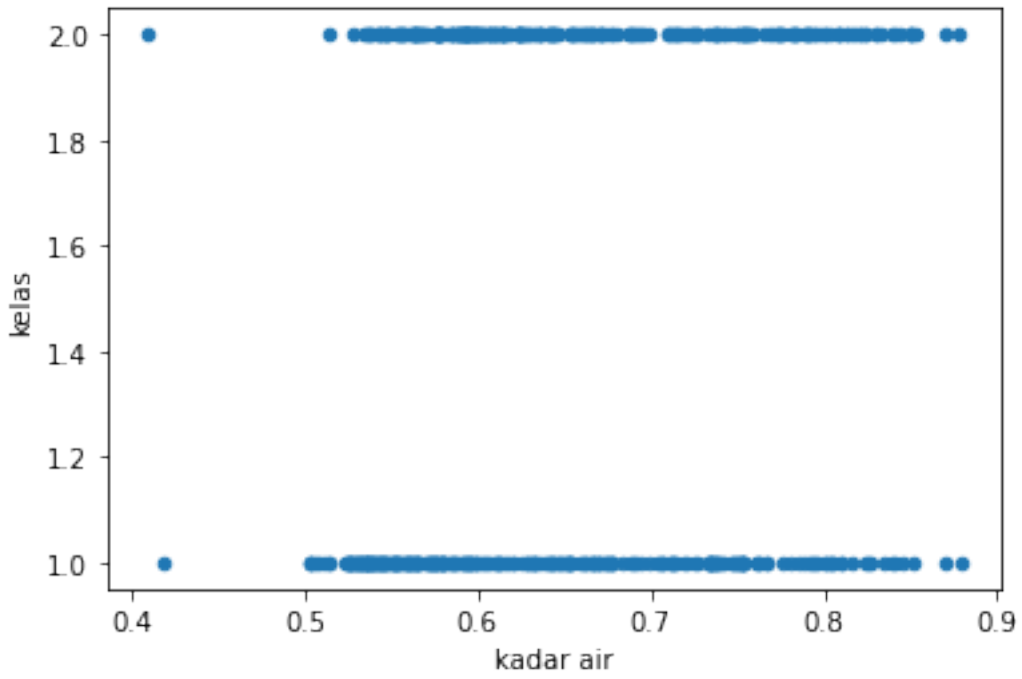


4.0.12 Kesimpulan :

4.0.13 Kadar Air dengan Kelas

[90]:
```
g.plot(kind='scatter', x='kadar air', y='kelas')
a = g['kadar air']
b = g['kelas']
a.corr(b)
```

[90]: 0.13434422605727642

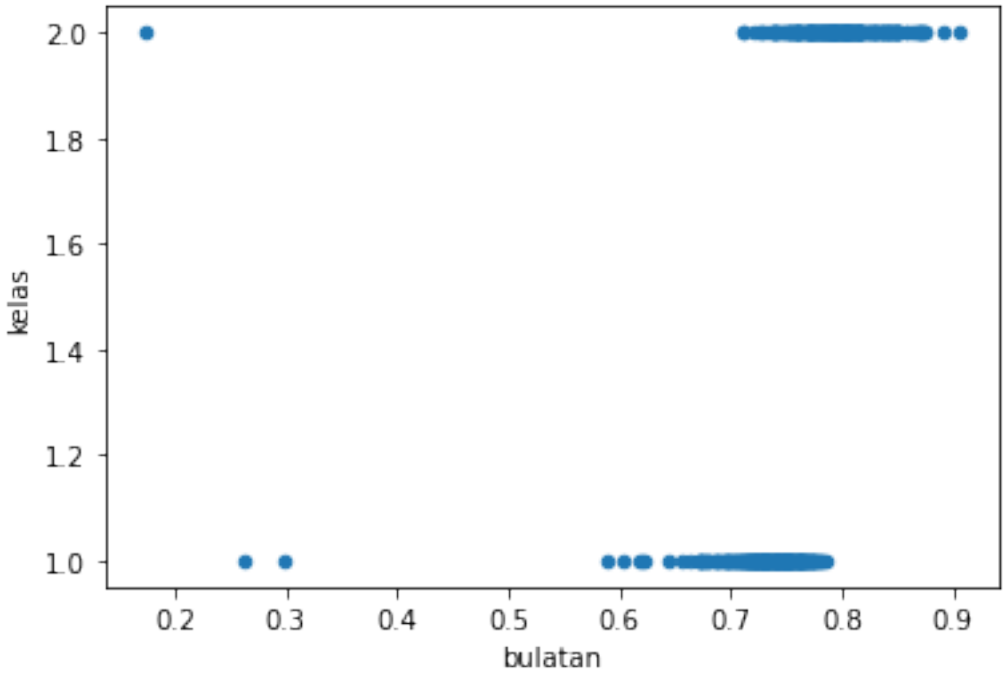


4.0.14 Kesimpulan :

4.0.15 Bulatan dengan Kelas

[91]:
```
g.plot(kind='scatter', x='bulatan', y='kelas')
a = g['bulatan']
b = g['kelas']
a.corr(b)
```

[91]: 0.5450045317240076

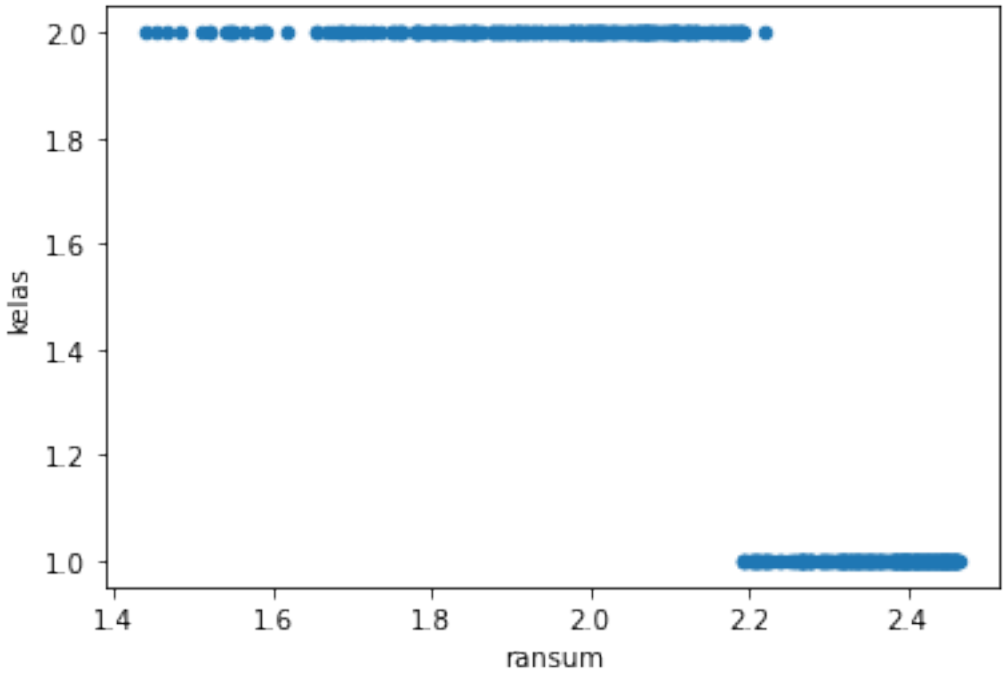


4.0.16 Kesimpulan :

4.0.17 Ransum dengan Kelas

[92]:
```
g.plot(kind='scatter', x='ransum', y='kelas')
a = g['ransum']
b = g['kelas']
a.corr(b)
```

[92]: -0.8399038681287493



4.0.18 Kesimpulan :