MODUL 2

Membangun Aplikasi Simulasi Quick Count Untuk Menduga Parameter Proporsi Pada Kasus Pilkada/Pilpres

(Minggu 2-3)

1. TUJUAN

Tujuan praktikum ini adalah mahasiswa mampu:

1) menerapkan beberapa random sampling;
2) menerapkan beberapa metode multistage sampling;
3) membuat algoritma sistem quick count pilkada;
4) melakukan perbandingan dan evaluasi estimator (parameter proporsi) dari beberapa metode sampling; dan
5) membangun aplikasi simulasi Quick Count Pilkada dengan bahasa pemrograman R/Jupyter Notebook atau bahasa pemrograman yang lain.

2. DASAR TEORI

Quick count atau penghitungan suara cepat adalah proses pencatatan hasil perolehan suara di ribuan TPS yang dipilih secara acak. Quick count adalah prediksi hasil pemilu berdasarkan fakta bukan berdasarkan opini. Pelaksanaan quick count ini menggunakan Kerangka Sampel (sampling frame) daftar desa atau kelurahan menurut Badan Pusat Statistik (BPS) dan daftar TPS pada PPS di desa atau kelurahan terpilih. Dengan quick count maka dapat diperkirakan perolehan suara pemilu atau pilkada secara cepat sehingga dapat memverifikasi hasil resmi penyelenggara pemilihan [1].

Faktor utama yang menyebabkan terjadinya perbedaan hasil quick count yang dilakukan lembaga-lebaga survel itu adalah karena penerapan jenis metode sampling (pengambilan sample dari populasi sebagai estimasi) yang berbeda-beda. Bagaimana sampel itu ditarik akan menentukan mana suara pemilih yang akan dipakai sebagai basis estimasi hasil pemilu. Sampel yang ditarik secara benar akan memberikan landasan kuat untuk mewakili karakteristik populasi (Scheaffer, 1990). Padahal ada beberapa teknik sampling yang digunakan pada pelaksanaan quick count, misalnya simple random

sampling, stratified random sampling, cluster random sampling, dan two-stage cluster sampling [2].

Metodologi statistik dan penarikan sampel yang ketat serta implementasi secara konsisten di lapangan dapat menghasilkan estimasi quick count yang akurat. Melakukan identifikasi terhadap berbagai faktor yang berdampak pada distribusi suara dalam populasi suara pemilih merupakan tolak ukur keberhasilan quick count. Apabila Pilkada dilakukan tanpa kecurangan, keakuratan quick count dapat dibandingkan dengan hasil resmi KPU. Sebaliknya jika Pilkada dilakukan penuh kecurangan, maka hasil quick count yang dapat dikatakan kredibel walaupun hasilnya berbeda dengan hasil resmi KPU. Selain dengan quick count biasanya pemantauan juga dilakukan dengan menggunakan metode penarikan sampel secara acak [3].

3. PERCOBAAN

Untuk membangun Sistem Quick Count Pilkada, lakukan beberapa percobaan berikut sebagai dasar atau bagian dari system tersebut.

3.1 Percobaan ke-1: Menaksir Parameter Proporsi dengan Random Sampling (without replacement)

Ilustrasikan ada 2 kandidat dalam sebuah kontes atau pemilihan pimpinan dengan 1 juta responden, maka ada beberapa kemungkinan, yaitu: 1) memilih kontestan ke-1; 2) memilih kontestan ke-2; 3) tidak memilih; dan 4) tidak sah. Jika tidak memilih atau tidak sah dinotasikan sebagai 0, memilih kontestan ke-1 dinotasikan 1, dan memilih kontestan ke-2 dinotasikan 2, maka langkah pertama sebelum mengestimasi parameter harus menghitung parameter proporsi terlebih dahulu, dengan langkah-langkah berikut:

1. Bangkitkan 1.000.000 bilangan random berdistribusi integer (0,2)
2. Lakukan counting dengan menghitung hasil yang bernilai 0, 1, dan 2.
3. Hitung nilai parameter proporsi.

Contoh program dalam python sebagai berikut:

```
import numpy as np
import random
y = np.zeros(1000000)
k=0
k0=0
k1=0
k2=0
n=len(y)
```

```
for i in range(n):
        y[k] = random.randint(0, 2)
        if(y[k]==0):
            k0=k0+1
        elif(y[k]==1):
            k1=k1+1
        else:
            k2=k2+1
        k=k+1
p0=round(100*(k0/n),2)
p1=round(100*(k1/n),2)
p2=round(100*(k2/n),2)
print("------Hasil Pemilihan (Parameter Proporsi)-------")
print("Kontestan - 1 = ", p1)
print("Kontestan - 2 = ", p2)
print("Abstain/Tidak Sah = ", p0)
```

Output:

```
------Hasil Pemilihan (Parameter Proporsi)-------
Kontestan - 1 =  33.38
Kontestan - 2 =  33.28
Abstain/Tidak Sah =  33.34
```

Selanjutnya, lakukan pendugaan parameter proporsi dari kasus tersebut menggunakan random sampling (without replacement/tanpa pengembalian) dengan ukuran sampel 10% dari populasi. Langkah-langkahnya adalah sebagai berikut:

1. Pilih secara acak data dari populasi sebanyak 10% atau 100.000.
2. Lakukan counting dengan menghitung hasil yang bernilai 0, 1, dan 2.
3. Hitung nilai estimasi parameter proporsi.

```
print("---Sampling without replacement:")
list_y=y.tolist()
n1=int(0.1*n)
sy = random.sample(list_y, n1)
#print(sy)
h=np.array(sy)
k=0
k0=0
k1=0
k2=0
for i in range(n1):
        if(h[k]==0):
            k0=k0+1
        elif(h[k]==1):
            k1=k1+1
        else:
```

```
            k2=k2+1
        k=k+1
rp0=round(100*(k0/n1),2)
rp1=round(100*(k1/n1),2)
rp2=round(100*(k2/n1),2)
print("------Hasil Pemilihan (Estimasi Parameter Proporsi)--
-----")
print("Kontestan - 1 = ", rp1)
print("Kontestan - 2 = ", rp2)
print("Abstain/Tidak Sah = ", rp0)
```

Output:

```
---Sampling without replacement:
------Hasil Pemilihan (Estimasi Parameter Proporsi)-------
Kontestan - 1 =  33.33
Kontestan - 2 =  33.36
Abstain/Tidak Sah =  33.31
```

Berikutnya dicoba untuk ukuran sampel mulai dari 5% sampai dengan 100% dari populasi.

```
print("---Sampling without replacement (n = 5% sd 95%):")
y = np.zeros(1000000)
rs0 = np.zeros(19)
rs1 = np.zeros(19)
rs2 = np.zeros(19)

for i in range(19):
    n1=int(0.05*(i+1)*n)
    sy = random.sample(list_y, n1)
    h=np.array(sy)
    k=0
    k0=0
    k1=0
    k2=0
    for j in range(n1):
        if(h[k]==0):
            k0=k0+1
        elif(h[k]==1):
            k1=k1+1
        else:
            k2=k2+1
        k=k+1
    rs0[i]=round(100*(k0/n1),2)
    rs1[i]=round(100*(k1/n1),2)
    rs2[i]=round(100*(k2/n1),2)
print("------Estimasi Parameter Proporsi (ukuran sampel 5% sd
95%)-------")
print("====================================")
print(" Sampel   K1    K2  Abstain/Tdk Sah")
```

```
print("====================================")
for i in range(19):
    print("   ",round(5*(i+1),0),"%      ",round(rs1[i],2),"
",round(rs2[i],2),"  ",round(rs0[i],2))
print("====================================")

#Menghitung persentase error parameter proporsi dengan hasil
estimasi
print("\n------Persentase Error-------")
print("================================================")
print(" Sampel  Error K1  Error K2  Error Abst/Tdk Sah")
print("================================================")
for i in range(19):
    print(" ",round(5*(i+1),0),"%    ",round(100*abs((rs1[i]-
p1)/p1),2),"            ",round(100*abs((rs2[i]-p2)/p2),2),"
",round(100*abs((rs0[i]-p0)/p0),2))
print("================================================")
```

Output:

```
------Estimasi Parameter Proporsi (ukuran sampel 5% sd 95%)---
----
====================================
 Sampel    K1      K2   Abstain/Tdk Sah
====================================
  5 %     33.52     33.05     33.43
  10 %    33.18     33.57     33.25
  15 %    33.04     33.49     33.47
  20 %    33.3      33.45     33.25
  25 %    33.36     33.36     33.28
  30 %    33.39     33.27     33.34
  35 %    33.34     33.28     33.39
  40 %    33.31     33.33     33.36
  45 %    33.3      33.25     33.45
  50 %    33.44     33.28     33.28
  55 %    33.4      33.32     33.29
  60 %    33.39     33.33     33.28
  65 %    33.35     33.3      33.35
  70 %    33.37     33.29     33.35
  75 %    33.38     33.28     33.34
  80 %    33.34     33.32     33.34
  85 %    33.38     33.32     33.3
  90 %    33.39     33.29     33.32
  95 %    33.38     33.27     33.34
====================================

------Persentase Error-------
================================================
 Sampel  Error K1  Error K2  Error Abst/Tdk Sah
================================================
  5 %       0.42      0.69      0.27
  10 %      0.6       0.87      0.27
```

```
  15 %      1.02       0.63       0.39
  20 %      0.24       0.51       0.27
  25 %      0.06       0.24       0.18
  30 %      0.03       0.03       0.0
  35 %      0.12       0.0        0.15
  40 %      0.21       0.15       0.06
  45 %      0.24       0.09       0.33
  50 %      0.18       0.0        0.18
  55 %      0.06       0.12       0.15
  60 %      0.03       0.15       0.18
  65 %      0.09       0.06       0.03
  70 %      0.03       0.03       0.03
  75 %      0.0        0.0        0.0
  80 %      0.12       0.12       0.0
  85 %      0.0        0.12        0.12
  90 %      0.03       0.03       0.06
  95 %      0.0        0.03       0.0
================================================
```

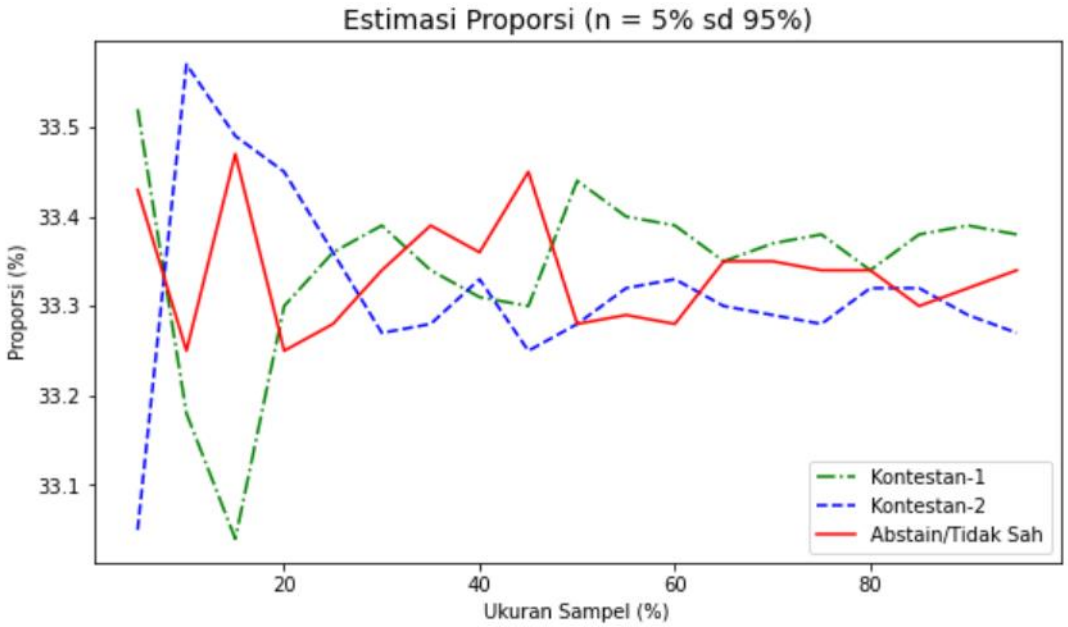
Tampilan dalam grafik.

```
# Display Estimasi Proporsi
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(9,5))
x = np.arange(5,100,5)

plt.plot(x,       rs1,       label       =       'Kontestan-1',
color='green',linestyle="-.")
plt.plot(x,       rs2,       label       =       'Kontestan-2',
color='blue',linestyle="--")
plt.plot(x,    rs0,    label    =    'Abstain/Tidak    Sah',
color='red',linestyle="-")


plt.xlabel('Ukuran Sampel (%)')
plt.title('Estimasi Proporsi (n = 5% sd 95%)',size = 14)
plt.ylabel('Proporsi (%)')
plt.legend(loc='lower right')
plt.savefig('c:/tugas_sdt/QC_random.png',dpi=100)
plt.show()
```



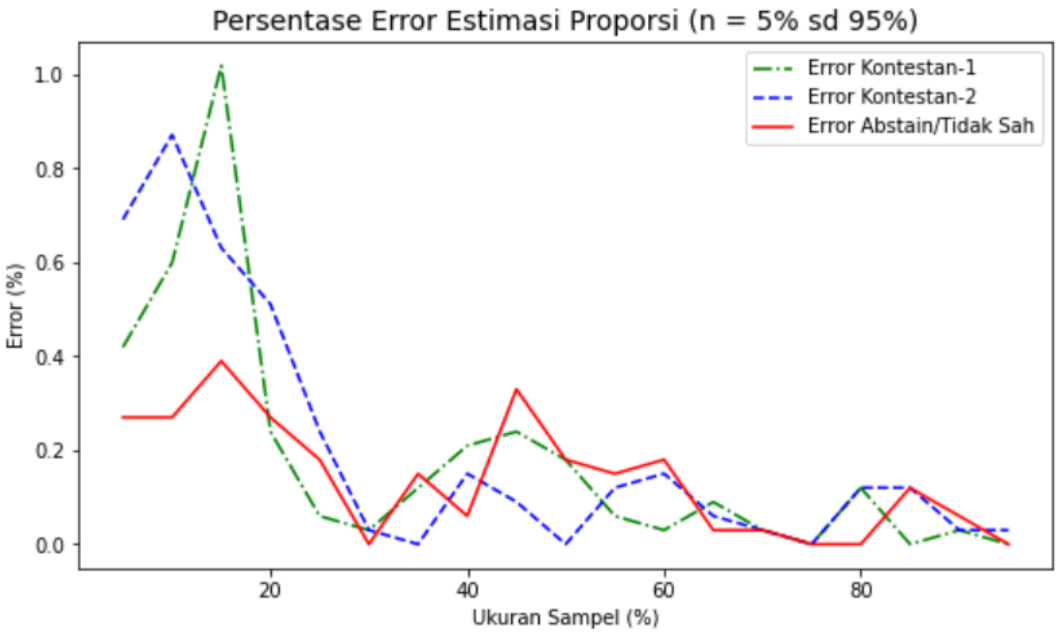
Tampilan dalam grafik untuk persentase error.

```
# Display Estimasi Persentase Error Proporsi
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(9,5))
x = np.arange(5,100,5)

plt.plot(x, abs(100*(rs1-p1)/p1), label = 'Error Kontestan-
1', color='green',linestyle="-.")
plt.plot(x, abs(100*(rs2-p2)/p2), label = 'Error Kontestan-
2', color='blue',linestyle="--")
plt.plot(x,    abs(100*(rs0-p0)/p0),    label    =    'Error
Abstain/Tidak Sah', color='red',linestyle="-")


plt.xlabel('Ukuran Sampel (%)')
plt.title('Persentase Error Estimasi Proporsi (n = 5% sd
95%)',size = 14)
plt.ylabel('Error (%)')
plt.legend(loc='upper right')
plt.savefig('c:/tugas_sdt/QC_error_random.png',dpi=100)
plt.show()
```



3.2 Percobaan ke-2: Menaksir Parameter Proporsi dengan Random Sampling (with replacement)

Untuk estimasi parameter proporsi dengan random sampling (with replacement/dengan pengembalian) menggunakan sintaksis random.choices(), mengikuti percobaan berikut.

```
cy = random.choices(list_y, k=n1)
#print(sy)
g=np.array(cy)
k=0
k0=0
k1=0
k2=0
n1=int(0.1*n)
for i in range(n1):
        if(g[k]==0):
            k0=k0+1
        elif(g[k]==1):
            k1=k1+1
        else:
            k2=k2+1
        k=k+1
wp0=round(100*(k0/n1),2)
wp1=round(100*(k1/n1),2)
wp2=round(100*(k2/n1),2)
print("Proporsi populasi: p0 =",p0,", p1 =",p1,", p2 =",p2)
print("------Estimasi Parameter Proporsi with replacement---
----")
```

```
print("Kontestan - 1 = ", wp1)
print("Kontestan - 2 = ", wp2)
print("Abstain/Tidak Sah = ", wp0) print("---Estimasi Proporsi
dengan random sampling with replacement:")
print("Estimasi Proporsi: p0 =",cp0,", p1 =",cp1,", p2 =",cp2)
```

Output:

```
Proporsi populasi: p0 = 33.34 , p1 = 33.38 , p2 = 33.28
------Estimasi Parameter Proporsi with replacement-------
Kontestan - 1 =  33.28
Kontestan - 2 =  33.22
Abstain/Tidak Sah =  33.5
```

3.3 Percobaan ke-3: Menaksir Parameter Proporsi dengan Cluster Sampling

Ilustrasikan ada 8 kelurahan dalam suatu kecamatan. Pada saat pemilihan Kepala Daerah, ingin diestimasi perolehan suara tiga kandidat Kepala Daerah. File QC1.csv adalah data sintetis sebanyak 2473 pemilih. Jika diasumsikan semua pemilih memilih salah satu kandidat (tidak ada abstain/suara tidak sah). Langkah-langkah percobaan adalah sebagai berikut.

1. Membaca data sistetis QC1.csv

```
import pandas as pd
import numpy as np
df = pd.read_csv('c:/tugas_sdt/QC1.csv')
df.head()
```

	No	Kelurahan	TPS	Pil
0	1	1	1	2
1	2	1	1	1
2	3	1	1	2
3	4	1	1	3
4	5	1	1	3

2. Menghitung jumlah pemilih di tiap kelurahan.

```
# Menghitung jumlah pemilih di tiap kelurahan
jumlah=df.groupby(['Kelurahan']).count()
jumlah=jumlah['Pil']
jumlah
```

```
Kelurahan
1    407
2    230
3    217
4    433
5    179
6    565
7    375
8     67
Name: Pil, dtype: int64
```

3. Menghitung pemilih di tiap TPS per kelurahan.

```
# Menghitung jumlah pemilih di tiap kelurahan
jumlah=df.groupby(['Kelurahan']).count()
jumlah=jumlah['Pil']
jumlah
```

```
Kelurahan  TPS
1          1      91
           2     111
           3     106
           4      73
           5      26
2          1      33
           2      34
           3     132
           4      31
3          1      32
           2      38
           3      99
           4      48
4          1      26
           2      28
           3      48
           4      61
           5      53
           6      97
           7     120
```

```
5          1       54
           2       34
           3       50
           4       41
6          1       24
           2       38
           3      101
           4      175
           5      183
           6       44
7          1       53
           2      126
           3       48
           4       66
           5       82
8          1       29
           2       38
Name: Pil, dtype: int64
```

4. Estimasi parameter proporsi (perolehan suara) masing-masing kandidat.

```
#Estimasi Perolehan Suara Kandidat-1 sd Kandidat-3
df1=100*df.groupby(['Kelurahan','Pil']).count()/df.groupb
y(['Kelurahan']).count()
df2=df1['TPS'].groupby(['Pil']).mean()
par = df2.to_numpy()
par
```

Output:

```
array([33.94504907, 32.66512331, 33.38982762])
```

5. Mengambil sampel menggunakan metode *cluster random sampling* (*cluster sampling*)

```
#Cluster Sampling
#Memilih secara random 4 dari 8 kelurahan
clusters                                                =
np.random.choice(np.arange(1,9),size=4,replace=False)

#Mendefinisikan sampel sebagai semua anggota dari 4
kelurahan yang terpilih
cluster_sample = df[df['Kelurahan'].isin(clusters)]

#Menyimpan data cluster terpilih ke dalam file xlsx
dfc=cluster_sample
dfc.to_excel("c:/tugas_sdt/QC1_clus.xlsx")
dfc[['Kelurahan','TPS','Pil']]
```

	Kelurahan	TPS	Pil
637	7	1	1
638	7	1	2
639	7	1	1
640	7	1	2
641	7	1	2
...	...	...	...
1724	4	7	1
1725	4	7	1
1726	4	7	2
1727	4	7	3
1728	4	7	1

1092 rows × 3 columns

6. Estimasi parameter proporsi, estimasi parameter dan simpangan baku

```
#Estimasi parameter proporsi
dfc1=100*dfc.groupby(['Kelurahan','Pil']).count()/dfc.groupby(
['Kelurahan']).count()
dmean=dfc1['TPS'].groupby(['Pil']).mean()
dstd = dfc1['TPS'].groupby(['Pil']).std()
m_clus = dmean.to_numpy()
s_clus = dstd.to_numpy()
p_error=100*(abs(par-m_clus)/par)
#Menampilkan parameter, estimasi parameter, simpangan baku,
persentasi error
par,m_clus,s_clus,p_error
```

Output:

```
(array([33.94504907, 32.66512331, 33.38982762]),
 array([33.30375492, 32.46598243, 34.23026265]),
 array([1.65929771, 1.82482979, 1.54381598]),
 array([1.8892126 , 0.60964373, 2.51703911]))
```

3.4 Percobaan ke-4: Menaksir Parameter Proporsi dengan Multistage Random Sampling

Pada percobaan ini mengikuti Langkah-langkah sebagai berikut:

1. Membaca data asli (QC1.csv)

```
#Membaca Data
df = pd.read_csv('c:/tugas_sdt/QC1.csv')
df.head()
```

	No	Kelurahan	TPS	Pil
0	1	1	1	2
1	2	1	1	1
2	3	1	1	2
3	4	1	1	3
4	5	1	1	3

2. Multistage Random Sampling

```
print("---Multistage Random Sampling:")
rs0 = np.zeros(19)
rs1 = np.zeros(19)
rs2 = np.zeros(19)
N=len(df)
#Stage-1: Random sampling
n = int(0.05*N)
sy = df.sample(n)
#Stage-2: Cluster Random Sampling, misal terpilih Kelurahan
= 2, 4, 5, 7
dfc = df[(df.Kelurahan == 2) | (df.Kelurahan == 4) |
(df.Kelurahan == 5) | (df.Kelurahan == 7)]
dfc.to_excel("c:/tugas_sdt/data_multistage.xlsx")
dfc
```

---Multistage Random Sampling:

	No	Kelurahan	TPS	Pil
407	408	2	1	1
408	409	2	1	3
409	410	2	1	2
410	411	2	1	2
411	412	2	1	1
...	...	...	...	...
1903	1904	5	4	1
1904	1905	5	4	1
1905	1906	5	4	3
1906	1907	5	4	3

3. Estimasi Persentase Suara masing-masing kandidat (Proporsi).

```
dfc1=100*dfc.groupby(['Kelurahan','Pil']).count()/dfc.gro
upby(['Kelurahan']).count()
dmean=dfc1['TPS'].groupby(['Pil']).mean()
dstd = dfc1['TPS'].groupby(['Pil']).std()
m_clus = dmean.to_numpy()
s_clus = dstd.to_numpy()
p_error=100*(abs(par-m_clus)/par)
#Menampilkan parameter, estimasi parameter, simpangan baku,
persentasi error
par,m_clus,s_clus,p_error
```

Output:

```
(array([33.94504907, 32.66512331, 33.38982762]),
 array([34.11963718, 31.5466033 , 34.33375953]),
 array([0.45112918, 1.22649561, 1.67761989]),
 array([0.5143257 , 3.42420263, 2.82700441]))
```

Referensi

_____, Memahami Metode Quick count. [1]https://news.detik.com/berita/d-815022/memahami-metode-quick-count, diakses tanggal 5 Juni 2022

[2] Scheaffer Richard L et all. (2012). Elementary Survey Sampling, Seventh Edition, Richard Stratton

Kismiantini (2007), Pengumpulan Data dengan Quick Count dan Exit Poll

4. LAPORAN RESMI

Ada dua opsi dalam proyek modul ini, yaitu Proyek – 1 dan Proyek – 2, masing-masing tim dapat memilih salah satu.

1) **Proyek – 1** merupakan kerja tim. Buat sistem/aplikasi Quick Count Pemilihan Kepala Daerah (Pilkada) Bukapeti/Wali Kota dengan data pembangkitan menggunakan Jupyter Python, dengan beberapa rule sebagai berikut.
   a. Kandidat Kepala Daerah sebanyak 3 orang.
   b. Golput, abstain dan suara tidak sah menjadi satu kelompok.
   c. Jumlah kecamatan antara 15 – 30 kecamatan.
   d. Jumlah desa/kelurahan di tiap kecamatan antara 10 – 15 desa/kelurahan.
   e. Jumlah TPS di tiap desa/kelurahan antara 4 – 7 TPS.
   f. Random sampling dan atau cluster random sampling diterapkan pada level TPS dan desa/kelurahan (multistage)
   g. Aplikasi dapat menggunakan console atau GUI.

2) **Proyek – 2** merupakan kerja tim. Buat sistem/aplikasi Quick Count Pemilihan Presiden (Pilpres) dengan data riil di laman https://pemilu2019.kpu.go.id/ menggunakan Jupyter Python, dengan beberapa rule sebagai berikut.
   a. Random sampling dan atau cluster random sampling diterapkan pada level TPS dan desa/kelurahan (multistage)
   b. Aplikasi dapat menggunakan console atau GUI.

Output minimal yang ditampilkan:
   a. Hitung parameter perolehan suara masing-masing kandidat.
   b. Hitung estimasi parameter dan selisih proporsi dengan parameter menggunakan metode random sampling dengan beberapa variasi ukuran sampel (5% sampai dengan 95%). Sajikan tabel dan grafiknya.
   c. Hitung estimasi parameter dengan cluster random sampling dengan beberapa variasi ukuran sampel (5% sampai dengan 95%). Sajikan tabel dan grafiknya.
   d. Hitung estimasi parameter dengan multistage random sampling
      - Kelompok dengan nomor ganjil:
        level TPS: cluster random sampling, level desa/kelurahan: random sampling
      - Kelompok dengan nomor genap:
        level TPS: random sampling, level desa/kelurahan: cluster random sampling

      dengan beberapa variasi ukuran sampel (5% sampai dengan 95%). Sajikan tabel dan grafiknya.

Lakukan analisis setiap output dan buat kesimpulan.
Dokumen yang wajib dikumpulkan:

1. Laporan Proyek-1
2. Source Code
3. File Presentasi atau Poster