Praktikum Analisa Statistika Terapan

Aplikasi Quick Count

Dosen Pengampu:
Ronny Susetyoko, S.Si., M.Si.

Puput Ayu S
3322600004

Bayu Kurniawan
3322600019

Abdul Muffid
3322600021

Program Studi Sains Data Terapan

IMPLEMENTASI APLIKASI *QUICK COUNT* PILKADA BERBASIS *DASHBOARD* (STUDI KASUS KABUPATEN SIDOARJO)

I. LATAR BELAKANG

Era reformasi pada tahun 1998 merupakan era di mana demokrasi mulai berkembang secara pesat. Pemilu menjadi salah satu aspek utama dalam menjaga serta memperkuat demokrasi yang terus berkembang. Ditambah pertumbuhan teknologi informasi yang pesat di Indonesia juga memainkan peran penting. Dengan penetrasi internet yang semakin luas, aplikasi *quick count* dapat menjadi alat yang strategis untuk mengumpulkan, memproses, dan menyebarkan data hasil pemilu secara efisien. Hal tersebut menciptakan transparansi yang dibutuhkan dalam pemilu guna menjaga integritas dan kepercayaan masyarakat.

Pemilihan Kepala Daerah (Pilkada) merupakan sebuah agenda rutin yang diadakan setiap lima tahun sekali di Indonesia. Sampai saat ini, masih tak sedikit hasil perhitungan suara yang dilakukan secara manual. Setelah tahap pemungutan selesai, proses perhitungan suara mulai dilakukan di tiap-tiap Tempat Pemungutan Suara (TPS), kemudian hasil perhitungan suara tersebut dilanjutkan ke kelurahan lalu diteruskan ke kecamatan untuk direkap oleh lembaga yang berwenang. Biasanya, KPU memerlukan waktu yang cukup lama untuk melakukan rekapitulasi hasil pemilihan, bahkan dapat mencapai tiga minggu atau lebih. Hal ini dikarenakan proses pengumpulan data dilakukan dari seluruh wilayah pelaksana Pilkada yang luas. Proses perhitungan cepat (*quick count*) perlu dilakukan untuk mendapatkan hasil sementara dari pemilihan yang diselenggarakan jika melihat lamanya hasil rekapitulasi resmi oleh KPU [1].

Metode *quick count* menjadi populer sejak diberlakukannya pemilu secara langsung. Metode ini dapat menerapkan teknik *sampling* probabilitas sehingga hasilnya jauh lebih akurat dan dapat mencerminkan populasi secara tepat. Metode ini menjadi sebuah alternatif baru yang digemari oleh berbagai pihak yang memiliki kepentingan dalam persaingan politik, baik di tingkat nasional maupun di tingkat lokal. Dengan metode ini, pihak-pihak tersebut dapat memiliki data pembanding yang dapat digunakan untuk mendeteksi adanya kemungkinan kecurangan yang terjadi pada proses tabulasi suara. Dalam statistika, metode ini juga bukanlah suatu hal yang baru. Metode ini menjadi populer karena kemudahan dalam penerapannya, biaya yang relatif rendah, serta kemampuannya dalam memberikan data yang akurat dengan tingkat akurasi yang tinggi.

Pilkada kabupaten Sidoarjo tahun 2020 menjadi sorotan karena adanya aplikasi *quick count* yang digunakan untuk memperkirakan hasil Pilkada sebelum pengumuman resmi dari Komisi Pemilihan Umum (KPU). Aplikasi *quick count* ini menggunakan metode

random sampling berbasis *android* dan *SMS gateway* untuk mengumpulkan data suara dari TPS yang kemudian dihitung secara cepat dan akurat. Aplikasi *quick count* ini menjadi salah satu cara untuk memantau jalannya Pilkada dan memberikan informasi awal kepada masyarakat mengenai hasil Pilkada sebelum pengumuman resmi dari KPU [2].

Namun, penggunaan aplikasi *quick count* ini juga menimbulkan kontroversi karena adanya dugaan manipulasi data dan kecurangan dalam penghitungan suara. Oleh karena itu, KPU kabupaten Sidoarjo mengimbau masyarakat untuk tidak terlalu bergantung pada hasil *quick count* dan menunggu pengumuman resmi dari KPU. Meskipun demikian, penggunaan aplikasi *quick count* ini menjadi salah satu bentuk partisipasi politik masyarakat dalam memantau jalannya Pilkada dan memberikan informasi awal mengenai hasil Pilkada [3].

II. TUJUAN

1. Menerapkan algoritma pembangkitan data berdasarkan distribusi *dirichlet*.
2. Mengaplikasikan metode *simple random sampling* dalam pengambilan sampel.
3. Mengaplikasikan metode *cluster sampling* dalam pengambilan sampel.
4. Mengaplikasikan metode *multistage cluster sampling* dalam pengambilan sampel.
5. Membandingkan serta mengevaluasi estimator (parameter proporsi dan persentasenya) dari beberapa metode *sampling*.

III. DATASET

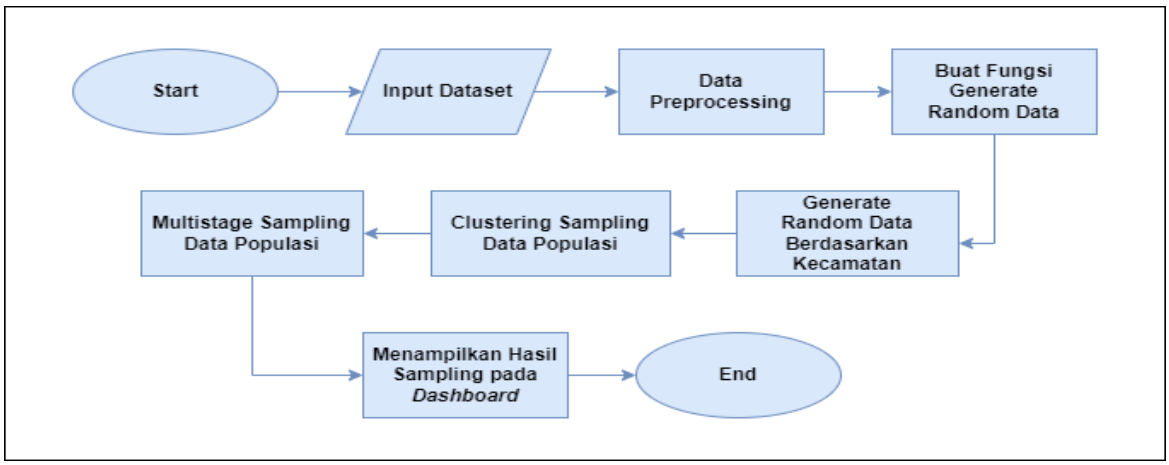
Sumber Data

Link Sumber : https://pilkada2020.kpu.go.id/#/pkwkk/tungsura/3515012019

Link Dataset : https://intip.in/DataPilkadaSidoarjo2020

Fitur : Tahun, Provinsi, Kabupaten, Kecamatan, dan Kelurahan.

IV. METODOLOGI



V. HASIL DAN PEMBAHASAN

A. Menghasilkan Data Acak dengan Aturan Tertentu

Langkah awal dalam proses ini adalah menghasilkan data acak terkait Pemilihan Kepala Daerah (Pilkada) di kabupaten Sidoarjo pada tahun 2020. Tujuannya adalah untuk menyiapkan data yang akan digunakan dalam proses *quick count*. Data yang dibutuhkan mencakup informasi mengenai berbagai tingkat wilayah, meliputi kecamatan, kelurahan, dan Tempat Pemungutan Suara (TPS). Dengan menghimpun data ini, pelaksanaan *quick count* dapat dilakukan dengan data yang komprehensif dan akurat sesuai kebutuhan.

Data yang telah didapatkan akan digunakan untuk menghasilkan data di TPS dengan menggunakan kode program sebagai berikut.

```
import pandas as pd
import random
import numpy as np
pd.set_option('display.max_rows', 10)
data = pd.read_csv("sidoarjo.csv", delimiter = ";")
data
```

```
     tahun            provinsi kabupaten    kecamatan           kelurahan
0     2020  PROVINSI JAWA TIMUR  SIDOARJO        TARIK         MLIRIPROWO
1     2020  PROVINSI JAWA TIMUR  SIDOARJO        TARIK        KEDUNGBOCOK
2     2020  PROVINSI JAWA TIMUR  SIDOARJO        TARIK         SINGOGALIH
3     2020  PROVINSI JAWA TIMUR  SIDOARJO        TARIK              TARIK
4     2020  PROVINSI JAWA TIMUR  SIDOARJO        TARIK         MERGOBENER
..     ...                   ...       ...          ...                ...
344   2020  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO        PENAMBANGAN
345   2020  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO          WARUBERON
346   2020  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO       BOGEMPINGGIR
347   2020  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO     KEDUNGSUKODANI
348   2020  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO  BAKUNGTEMENGGUNGAN
[349 rows x 5 columns]
```

Selanjutnya membuat aturan awal sesuai dengan hasil perhitungan hasil Pilkada di kabupaten Sidoarjo pada tahun 2020. Di mana data ini nantinya dapat digunakan untuk perhitungan hasil pemilihan, seperti persentase suara kandidat, jumlah suara sah, jumlah suara tidak sah, dan tingkat partisipasi pemilih di setiap kecamatan atau kelurahan dalam kabupaten Sidoarjo.

```
suara_1 = 373_673
suara_2 = 387_688
suara_3 = 212_977
golput = 4_257
```

```
total_tps_tarik = 112
total_tps_jprambon = 139
total_tps_krembung = 125
total_tps_porong = 124
total_tps_jabon = 99
total_tps_tanggulangin = 152
total_tps_candi = 261
total_tps_tulangan = 175
total_tps_wonoayu = 145
total_tps_sukodono = 208
total_tps_sidoarjo = 361
total_tps_buduran = 167
total_tps_sedati = 165
total_tps_waru = 363
total_tps_gedangan = 207
total_tps_taman = 365
total_tps_krian = 221
total_tps_balongbendo = 132
data
```

	tahun	provinsi	kabupaten	kecamatan	kelurahan
0	2020	PROVINSI JAWA TIMUR	SIDOARJO	TARIK	MLIRIPROWO
1	2020	PROVINSI JAWA TIMUR	SIDOARJO	TARIK	KEDUNGBOCOK
2	2020	PROVINSI JAWA TIMUR	SIDOARJO	TARIK	SINGOGALIH
3	2020	PROVINSI JAWA TIMUR	SIDOARJO	TARIK	TARIK
4	2020	PROVINSI JAWA TIMUR	SIDOARJO	TARIK	MERGOBENER
..	...	...	...	...	...
344	2020	PROVINSI JAWA TIMUR	SIDOARJO	BALONGBENDO	PENAMBANGAN
345	2020	PROVINSI JAWA TIMUR	SIDOARJO	BALONGBENDO	WARUBERON
346	2020	PROVINSI JAWA TIMUR	SIDOARJO	BALONGBENDO	BOGEMPINGGIR
347	2020	PROVINSI JAWA TIMUR	SIDOARJO	BALONGBENDO	KEDUNGSUKODANI
348	2020	PROVINSI JAWA TIMUR	SIDOARJO	BALONGBENDO	BAKUNGTEMENGGUNGAN

[349 rows x 5 columns]

Setelah menyiapkan *dataset* pada proses sebelumnya, langkah selanjutnya yaitu melakukan pengacakan data dengan memerhatikan bahwa total suara harus tetap dengan data asli yang ada. Pertama, menghapus kolom "**tahun**" karena semua data berada dalam tahun yang sama. Selanjutnya, memisahkan data berdasarkan kecamatan di kabupaten Sidoarjo, sehingga setiap kecamatan memiliki *dataframe* sendiri. Hal inimemungkinkan untuk melakukan analisis yang lebih terfokus pada hasil pemilihan di masing-masing kecamatan.

```
data = data.drop(columns = ["tahun"])
kab_adm_sidoarjo = data[data["kabupaten"] == "SIDOARJO"]
kec_adm_tarik = data[data["kecamatan"] == "TARIK"]
kec_adm_prambon = data[data["kecamatan"] == "PRAMBON"]
kec_adm_krembung = data[data["kecamatan"] == "KREMBUNG"]
kec_adm_porong = data[data["kecamatan"] == "PORONG"]
kec_adm_jabon = data[data["kecamatan"] == "JABON"]
kec_adm_tanggulangin = data[data["kecamatan"] == "TANGGULANGIN"]
kec_adm_candi = data[data["kecamatan"] == "CANDI"]
kec_adm_tulangan = data[data["kecamatan"] == "TULANGAN"]
kec_adm_wonoayu = data[data["kecamatan"] == "WONOAYU"]
kec_adm_sukodono = data[data["kecamatan"] == "SUKODONO"]
kec_adm_sidoarjo = data[data["kecamatan"] == "SIDOARJO"]
```

```
kec_adm_buduran = data[data["kecamatan"] == "BUDURAN"]
kec_adm_sedati = data[data["kecamatan"] == "SEDATI"]
kec_adm_waru = data[data["kecamatan"] == "WARU"]
kec_adm_gedangan = data[data["kecamatan"] == "GEDANGAN"]
kec_adm_krian = data[data["kecamatan"] == "KRIAN"]
kec_adm_balongbendo = data[data["kecamatan"] == "BALONGBENDO"]
```

Selanjutnya, membuat fungsi ***generate_tps()*** untuk menghasilkan data acak dengan memperhatikan total TPS di setiap kelurahan, dan menyesuaikan distribusi suara sesuai dengan data riil yang ada.

```
def generate_tps(data, suara_1, suara_2, suara_3, golput):
  kelurahan = data["kelurahan"].unique()
  kelurahan_tps = []
  for kel in kelurahan:
    kelurahan_tps.append({
      "kelurahan": kel,
      "jml_tps": random.randint(4, 7)
    })

  tps = []

  for kel in kelurahan_tps:
    for i in range(kel["jml_tps"]):
      tps.append({
          "kelurahan": kel["kelurahan"],
          "no_tps": i + 1,
          "suara_1": 0,
          "suara_2": 0,
          "suara_3": 0,
          "golput": 0
      })

  random_suara_1 = (np.random.dirichlet(np.ones(len(tps)),size = 1)[0]
* suara_1).tolist()
  random_suara_2 = (np.random.dirichlet(np.ones(len(tps)),size = 1)[0]
* suara_2).tolist()
  random_suara_3 = (np.random.dirichlet(np.ones(len(tps)),size = 1)[0]
* suara_3).tolist()
  random_golput = (np.random.dirichlet(np.ones(len(tps)),size = 1)[0]
* golput).tolist()

  random_suara_1 = [round(i) for i in random_suara_1]
  random_suara_2 = [round(i) for i in random_suara_2]
  random_suara_3 = [round(i) for i in random_suara_3]
  random_golput = [round(i) for i in random_golput]

  for i in range(len(tps)):
    tps[i]["suara_1"] = random_suara_1[i]
    tps[i]["suara_2"] = random_suara_2[i]
    tps[i]["suara_3"] = random_suara_3[i]
    tps[i]["golput"] = random_golput[i]

  tps.append({
    "kelurahan": "TOTAL",
    "no_tps": sum([kel["jml_tps"] for kel in kelurahan_tps]),
    "suara_1": sum([t["suara_1"] for t in tps]),
    "suara_2": sum([t["suara_2"] for t in tps]),
    "suara_3": sum([t["suara_3"] for t in tps]),
    "golput": sum([t["golput"] for t in tps])
```

```
  })

  return pd.DataFrame(tps)
```

Deskripsi fungsi *generate_tps()*:

Fungsi ini menerima lima parameter, yaitu ***data*** (data awal), ***suara_1*** (total suara kandidat 1), ***suara_2*** (total suara kandidat 2), ***suara_3*** (total suara kandidat 3), dan ***golput*** (total golput).

Pertama, fungsi ini mengumpulkan daftar kelurahan yang ada dalam *dataset* dan menentukan secara acak jumlah TPS di setiap kelurahan, dengan rentang antara 4 hingga 7 TPS per kelurahan. Selanjutnya, untuk setiap TPS yang ada, fungsi ini menginisialisasi nilai suara untuk setiap kandidat (***suara_1***, ***suara_2***, ***suara_3***) dan jumlah golput dengan nilai awal 0.

Selanjutnya, nilai suara untuk masing-masing TPS dihasilkan secara acak menggunakan distribusi *Dirichlet*, sehingga menjaga jumlah total suara sesuai dengan data riil yang ada. Kemudian, nilai suara dibulatkan dan disimpan dalam bentuk *list*. Hal tersebut bertujuan untuk membuat jumlah suara yang lebih realistis yang diterima oleh setiap kandidat dan jumlah golput sesuai dengan data yang sebenarnya.

Terakhir, fungsi ini menghitung total suara dan jumlah golput di seluruh TPS di semua kelurahan. Serta menambahkan baris "**TOTAL**" dalam *dataframe* untuk merekap hasil keseluruhan. Hasil dari fungsi ini adalah sebuah *dataframe* yang berisi data acak mengenai suara kandidat dan golput di setiap TPS di semua kelurahan, dengan total yang sesuai dengan data riil yang ada. Di mana data ini dapat digunakan untuk melakukan analisis lebih lanjut terkait hasil pemilihan dengan mempertimbangkan variasi di tingkat TPS dan kelurahan.

Dilakukan pengacakan data yang dilakukan dengan fungsi ***generate_tps()*** untuk menghasilkan suara secara acak untuk setiap TPS di berbagai kelurahan. Selanjutnya, juga dilakukan perhitungan total suara untuk setiap kandidat dan jumlah golput guna memberikan gambaran keseluruhan tentang hasil pemilihan di kabupaten Sidoarjo. Lalu, data hasil pemilihan yang dihasilkan secara acak kemudian digabungkan dengan data asli. Hal ini bertujuan untuk membandingkan hasil pemilihan yang dihasilkan secara acak dengan data riil yang ada.

```
generate = generate_tps(data, suara_1, suara_2, suara_3, golput)
generate

total_row = pd.DataFrame([{
  "provinsi": "TOTAL",
  "kabupaten": len(data["kabupaten"].unique()),
```

```
  "kecamatan": len(data["kecamatan"].unique()),
  "kelurahan": len(data["kelurahan"].unique()),
  "no_tps": len(generate["kelurahan"].unique()) * len(generate["no_tps
"].unique()),
  "suara_1": generate["suara_1"].sum(),
  "suara_2": generate["suara_2"].sum(),
  "suara_3": generate["suara_3"].sum(),
  "golput": generate["golput"].sum()
}])

completed_data= pd.merge(data, generate, on = "kelurahan", how = "oute
r")
completed_data
```

```
                 provinsi kabupaten    kecamatan           kelurahan  no_tps
\
0     PROVINSI JAWA TIMUR  SIDOARJO        TARIK          MLIRIPROWO       1
1     PROVINSI JAWA TIMUR  SIDOARJO        TARIK          MLIRIPROWO       2
2     PROVINSI JAWA TIMUR  SIDOARJO        TARIK          MLIRIPROWO       3
3     PROVINSI JAWA TIMUR  SIDOARJO        TARIK          MLIRIPROWO       4
4     PROVINSI JAWA TIMUR  SIDOARJO        TARIK         KEDUNGBOCOK       1
...                   ...       ...          ...                 ...     ...
1929  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO  BAKUNGTEMENGGUNGAN       3
1930  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO  BAKUNGTEMENGGUNGAN       4
1931  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO  BAKUNGTEMENGGUNGAN       5
1932  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO  BAKUNGTEMENGGUNGAN       6
1933                   NaN       NaN          NaN               TOTAL    1861

      suara_1  suara_2  suara_3  golput
0          52      173       49       4
1          14      206      148       0
2         265       38       14       2
3         127      320      224       1
4          76      131       32       2
...       ...      ...      ...     ...
1929      823        2       51       1
1930      838      135       73       1
1931      152      104       96       0
1932        7       58        9       2
1933   373672   387715   212991    4229

[1934 rows x 9 columns]
```

Hasil penggabungan (*merge*) antara data asli dan data yang dihasilkan secara acak yang disimpan dalam variabel ***completed_data*** merupakan *dataframe* yang mencakup seluruh informasi tentang pemilihan, meliputi total suara kandidat dan golput di setiap TPS.

B. Menaksir Estimasi Parameter Menggunakan Metode *Random Sampling*

Menghasilkan data suara acak di kabupaten Sidoarjo dengan mempertimbangkan jumlah suara kandidat dan golput yang telah ditentukan sebelumnya. Data ini mencakup setiap TPS di masing-masing kelurahan.

```
# Data TPS beserta suara di Kab Sidoarjo
suara_1 = 373673
suara_2 = 387688
suara_3 = 212977
golput = 4257
data_suara_kab = generate_tps(kab_adm_sidoarjo, suara_1, suara_2, suar
```

```
a_3, golput)
data_suara_kab
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	MLIRIPROWO	1	91	83	292	5
1	MLIRIPROWO	2	150	275	55	2
2	MLIRIPROWO	3	311	175	261	5
3	MLIRIPROWO	4	188	152	255	4
4	MLIRIPROWO	5	79	45	64	2
...	...	...	...	...	...	...
1857	BAKUNGTEMENGGUNGAN	2	60	56	195	4
1858	BAKUNGTEMENGGUNGAN	3	321	373	183	1
1859	BAKUNGTEMENGGUNGAN	4	126	990	99	1
1860	BAKUNGTEMENGGUNGAN	5	350	134	3	10
1861	TOTAL	1861	373657	387671	212965	4221

[1862 rows x 6 columns]

Pada *output*, dapat dilihat suara untuk setiap TPS, jumlah suara kandidat, dan jumlah golput, serta total keseluruhan suara pada tingkat kabupaten.

Kemudian, mengumpulkan data suara dari berbagai kelurahan di kabupaten SIdoarjo, menghapus baris total dari setiap data, dan menggabungkan data suara dengan data asli kabupaten.

```
# collect all data suara and remove total row from every data
data_suara = pd.concat([data_suara_kab])
data_suara = data_suara[data_suara["kelurahan"] != "TOTAL"]

total_row = pd.DataFrame([{
  "kabupaten": "TOTAL",
  "kecamatan": len(data["kecamatan"].unique()),
  "kelurahan": len(data_suara["kelurahan"].unique()),
  "no_tps": len(data_suara["kelurahan"].unique()) * len(data_suara["no
_tps"].unique()),
  "suara_1": data_suara["suara_1"].sum(),
  "suara_2": data_suara["suara_2"].sum(),
  "suara_3": data_suara["suara_3"].sum(),
  "golput": data_suara["golput"].sum()
}])

# concat data suara with data
completed_data_kab = pd.merge(data, data_suara, on = "kelurahan", how
= "outer")
completed_data_kab = pd.concat([completed_data_kab, total_row], ignore
_index = True)
```

Manfaat dari program di atas adalah memastikan data suara yang lengkap dan terperinci untuk memberikan gambaran yang lebih komprehensif mengenai hasil pemilihan di seluruh bagian di kabupaten.

Membangkitkan data acak di setiap kecamatan dengan menggunakan fungsi ***generate_tps()***.

```
# Data TPS beserta suara di Kec Tarik
suara_1 = 20409
suara_2 = 7850
suara_3 = 10789
golput = 203
data_suara_tarik = generate_tps(kec_adm_tarik, suara_1, suara_2, suara
_3, golput)
data_suara_tarik
```

```
       kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0     MLIRIPROWO       1       35       72      104       0
1     MLIRIPROWO       2      221       12       57       0
2     MLIRIPROWO       3       92       14      158       2
3     MLIRIPROWO       4       38        2       16       8
4    KEDUNGBOCOK       1      300       36       97       0
..           ...     ...      ...      ...      ...     ...
104    KEDINDING       1      299      102      227       8
105    KEDINDING       2      325      230      690       0
106    KEDINDING       3       32        1      206       4
107    KEDINDING       4      118       23      104       1
108        TOTAL     108    20412     7851    10791     198

[109 rows x 6 columns]
```

```
# Data TPS beserta suara di Kec Prambon
suara_1 = 14442
suara_2 = 18059
suara_3 = 11328
golput = 186
data_suara_prambon = generate_tps(kec_adm_prambon, suara_1, suara_2, s
uara_3, golput)
data_suara_prambon
```

```
     kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0      PRAMBON       1       15      336      349       2
1      PRAMBON       2      233       34      207       1
2      PRAMBON       3       42      104      157       3
3      PRAMBON       4      163       14       17       4
4      PRAMBON       5      111      181       35       0
..         ...     ...      ...      ...      ...     ...
110  WATUTULIS       2      474      225      275       1
111  WATUTULIS       3       44      342      250       2
112  WATUTULIS       4       74      134       40       0
113  WATUTULIS       5      526      156       25       0
114      TOTAL     114    14446    18062    11325     180

[115 rows x 6 columns]
```

```
# Data TPS beserta suara di Kec Krembung
suara_1 = 14118
suara_2 = 18025
suara_3 = 8905
golput = 163
data_suara_krembung = generate_tps(kec_adm_krembung, suara_1, suara_2,
suara_3, golput)
data_suara_krembung
```

```
       kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0     TAMBAKREJO       1       69        1        1       2
```

```
1     TAMBAKREJO       2       39       48      119       1
2     TAMBAKREJO       3      160      279       42       2
3     TAMBAKREJO       4       13      111       59       1
4     TAMBAKREJO       5      141      437       94       2
..           ...     ...      ...      ...      ...     ...
101  BALONGGARUT       1      121       98       36       2
102  BALONGGARUT       2       34       64       99       2
103  BALONGGARUT       3       62       55        2       1
104  BALONGGARUT       4       60      122      139       0
105        TOTAL     105    14122    18028     8905     158

[106 rows x 6 columns]
```

```
# Data TPS beserta suara di Kec Porong
suara_1 = 12708
suara_2 = 14267
suara_3 = 8909
golput = 187
data_suara_porong = generate_tps(kec_adm_porong, suara_1, suara_2, sua
ra_3, golput)
data_suara_porong
```

```
    kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0      PORONG       1       21      138      131       2
1      PORONG       2       91       84       56       1
2      PORONG       3      100      156      306       2
3      PORONG       4      238       11       28       3
4      PORONG       5       53      128        1       1
..        ...     ...      ...      ...      ...     ...
82  PESAWAHAN       3      303      324       13       2
83  PESAWAHAN       4      165      108      124       1
84  PESAWAHAN       5        0      140       89       1
85  PESAWAHAN       6        0       37       75       6
86      TOTAL      86    12708    14268     8906     185

[87 rows x 6 columns]
```

```
# Data TPS beserta suara di Kec Jabon
suara_1 = 13805
suara_2 = 12243
suara_3 = 5720
golput = 176
data_suara_jabon = generate_tps(kec_adm_jabon, suara_1, suara_2, suara
_3, golput)
data_suara_jabon
```

```
     kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0     PANGGREH       1      176      286       27       3
1     PANGGREH       2       72       11       15       3
2     PANGGREH       3      151       79      215       1
3     PANGGREH       4       35       21       23       2
4   TROMPOASRI       1        5      196       30       4
..         ...     ...      ...      ...      ...     ...
85    PERMISAN       1       45      229       80       2
86    PERMISAN       2       59      288       49       2
87    PERMISAN       3       15      113       61       2
88    PERMISAN       4       61      123       14       2
89       TOTAL      89    13806    12244     5721     181
```

```
[90 rows x 6 columns]
```

```
# Data TPS beserta suara di Kec Tulangan
suara_1 = 16367
suara_2 = 28600
suara_3 = 10589
golput = 243
data_suara_tulangan = generate_tps(kec_adm_tulangan, suara_1, suara_2,
suara_3, golput)
data_suara_tulangan
```

```
    kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0       JANTI       1        5      246      303       1
1       JANTI       2      174      438      141       2
2       JANTI       3       47      125       86       6
3       JANTI       4      274      227        9       2
4       JANTI       5        7      187       24       6
..        ...     ...      ...      ...      ...     ...
119  GRABAGAN       4       83      638      141       3
120  GRABAGAN       5      507      271       32       2
121  GRABAGAN       6       81     1069        8       2
122  GRABAGAN       7       21      310      206       1
123     TOTAL     123    16366    28598    10585     242

[124 rows x 6 columns]
```

```
# Data TPS beserta suara di Kec Candi
suara_1 = 29845
suara_2 = 31612
suara_3 = 13356
golput = 265
data_suara_candi = generate_tps(kec_adm_candi, suara_1, suara_2, suara
_3, golput)
data_suara_candi
```

```
         kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0    KARANGTANJUNG       1        2      439      171       0
1    KARANGTANJUNG       2      134      251       47       1
2    KARANGTANJUNG       3       19      105       29       3
3    KARANGTANJUNG       4      212      325       33       3
4    KARANGTANJUNG       5      354      204        1       0
..             ...     ...      ...      ...      ...     ...
131       LARANGAN       2      121       42       25       2
132       LARANGAN       3       26      475       18       0
133       LARANGAN       4      178      543       41       0
134       LARANGAN       5       41       75       55       1
135          TOTAL     135    29842    31610    13352     260

[136 rows x 6 columns]
```

```
# Data TPS beserta suara di Kec Tanggulangin
suara_1 = 17873
suara_2 = 15280
suara_3 = 13643
golput = 221
```

```
data_suara_tanggulangin = generate_tps(kec_adm_tanggulangin, suara_1,
suara_2, suara_3, golput)
data_suara_tanggulangin
```

```
       kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0    KALISAMPURNO       1      377       76        3       1
1    KALISAMPURNO       2      125      756      135       1
2    KALISAMPURNO       3       91      132      133       0
3    KALISAMPURNO       4       70      159       37       0
4    KALISAMPURNO       5      359      193      108       2
..            ...     ...      ...      ...      ...     ...
92       RANDEGAN       2      387       44       86       0
93       RANDEGAN       3      214      194      298       5
94       RANDEGAN       4       85      140      355       5
95       RANDEGAN       5      214       31       12       2
96          TOTAL      96    17868    15283    13641     218

[97 rows x 6 columns]
```

```
# Data TPS beserta suara di Kec Sidoarjo
suara_1 = 35724
suara_2 = 29583
suara_3 = 19840
golput = 241
data_suara_sidoarjo = generate_tps(kec_adm_sidoarjo, suara_1, suara_2,
suara_3, golput)
data_suara_sidoarjo
```

```
    kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0     SIDOKARE       1      543       89       33       1
1     SIDOKARE       2     1034       25        7       1
2     SIDOKARE       3      211      310      267       2
3     SIDOKARE       4       73      115       20       0
4        CELEP       1       33      252       89       2
..         ...     ...      ...      ...      ...     ...
120    SUMPUT       2       17       88       63       2
121    SUMPUT       3      368      808       10       2
122    SUMPUT       4      493      128       27       2
123    SUMPUT       5      194      100      357       1
124     TOTAL     124    35723    29583    19842     243

[125 rows x 6 columns]
```

```
# Data TPS beserta suara di Kec Wonoayu
suara_1 = 20029
suara_2 = 16277
suara_3 = 9504
golput = 187
data_suara_wonoayu = generate_tps(kec_adm_wonoayu, suara_1, suara_2, s
uara_3, golput)
data_suara_wonoayu
```

```
        kelurahan  no_tps  suara_1  suara_2  suara_3  golput
0         TANGGUL       1      248       12      103       0
1         TANGGUL       2      127       59       43       3
2         TANGGUL       3      397       29       47       2
3         TANGGUL       4      101       11       54       0
4    SIMOKETAWANG       1      201        7       60       0
..            ...     ...      ...      ...      ...     ...
```

```
131    CANDINEGORO        4       420       205        95        1
132    CANDINEGORO        5        99        35       194        1
133    CANDINEGORO        6       689       403        73        1
134    CANDINEGORO        7        48       228        29        1
135          TOTAL      135     20025     16275      9502      179

[136 rows x 6 columns]
```

Hasil pembangkitan suara di setiap kecamatan menunjukkan adanya variasi yang signifikan. Terdapat perbedaan yang mencolok dalam jumlah total suara di setiap kecamatan, dengan beberapa kecamatan memiliki lebih banyak TPS daripada yang lain. Selain itu, distribusi suara antara kandidat juga beragam. Beberapa kecamatan menunjukkan preferensi pemilih yang hampir merata di antara ketiga kandidat, sementara yang lain memiliki perbedaan yang besar. Jumlah golput juga berbeda, yang mencerminkan tingkat partisipasi yang bervariasi di setiap kecamatan.

Selanjutnya, mengumpulkan dan mengintegrasikan data suara dari setiap kecamatan dalam satu kesatuan data yang lebih besar, serta menghapus baris "**TOTAL**" dari setiap data.

```
# collect all data suara and remove total row from every data
data_suara = pd.concat([data_suara_tarik, data_suara_prambon, data_sua
ra_krembung, data_suara_porong, data_suara_jabon,
                        data_suara_tulangan, data_suara_candi, data_su
ara_tanggulangin, data_suara_sidoarjo, data_suara_wonoayu])
data_suara = data_suara[data_suara["kelurahan"] != "TOTAL"]

total_row = pd.DataFrame([{
  "kabupaten": "TOTAL",
  "kecamatan": len(data["kecamatan"].unique()),
  "kelurahan": len(data_suara["kelurahan"].unique()),
  "no_tps": len(data_suara["kelurahan"].unique()) * len(data_suara["no
_tps"].unique()),
  "suara_1": data_suara["suara_1"].sum(),
  "suara_2": data_suara["suara_2"].sum(),
  "suara_3": data_suara["suara_3"].sum(),
  "golput": data_suara["golput"].sum()
}])

# concat data suara with data
completed_data_kec = pd.merge(data, data_suara, on = "kelurahan", how
= "outer")
completed_data_kec = pd.concat([completed_data_kec, total_row], ignore
_index = True)
```

Dalam pengembangan program *quick count*, digunakan pendekatan dengan menghasilkan data secara acak dan mengambil sampel secara acak. Pendekatan ini dianggap paling sesuai untuk menggambarkan sistem *quick count* yang sebenarnya. Di mana tahapan selanjutnya adalah melakukan pengambilan data dengan metode *random sampling* dan lain-lain untuk melakukan analisis lebih lanjut.

Estimasi parameter hasil *quick count* dilakukan dengan mengambil sampel data acak sebanyak 5% hingga 95%. Estimasi parameter ini mencakup proporsi suara untuk masing-masing kandidat (suara_1, suara_2, suara_3) dan jumlah golput dalam sampel yang diambil. Hasil estimasi parameter ini berguna untuk memberikan gambaran perkiraan hasil pemilihan berdasarkan sampel yang diambil, sehingga dapat digunakan sebagai referensi awal untuk memprediksi hasil pemilihan pada tingkat kecamatan. Semakin besar persentase sampel yang diambil, semakin mendekati estimasi hasil sebenarnya, tetapi dengan biaya pengambilan sampel yang lebih besar.

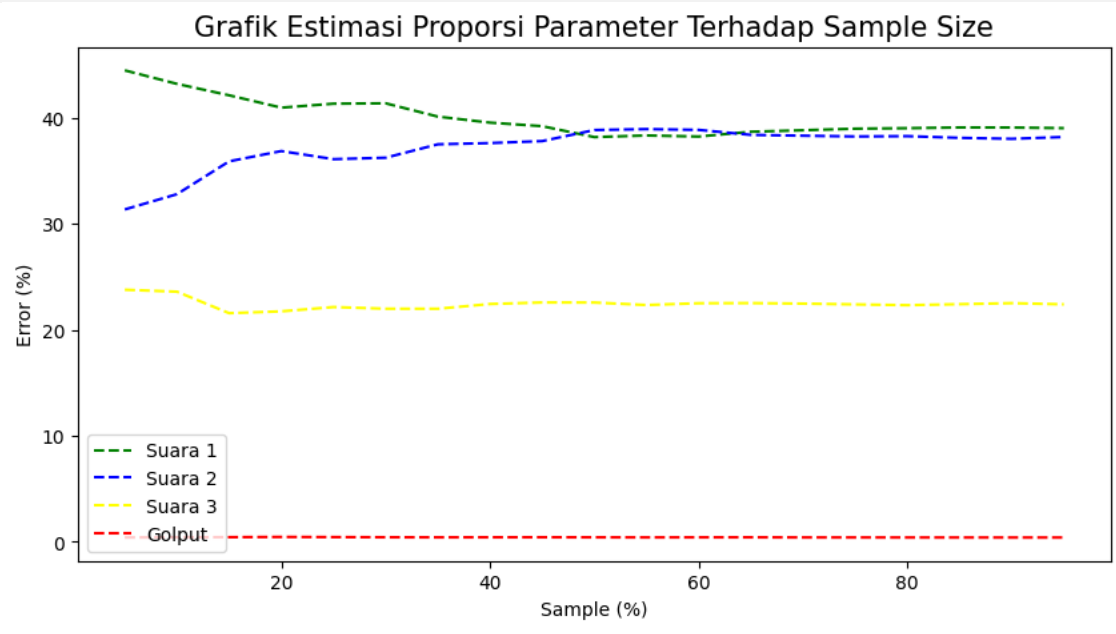
```
# estimasi parameter
estimation_parameter_data = []

for i in range(1, 20):
  random_sampling = completed_data_kec.sample(frac = i / 20, random_st
ate = 42).reset_index(drop = True)

  # sum all suara_1, suara_2, suata 3, and golput
  suara_1 = random_sampling["suara_1"].sum()
  suara_2 = random_sampling["suara_2"].sum()
  suara_3 = random_sampling["suara_3"].sum()
  golput = random_sampling["golput"].sum()

  # calculate percentage of suara_1, suara_2, suara 3, and golput
  percentage_suara_1 = (suara_1 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_suara_2 = (suara_2 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_suara_3 = (suara_3 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_golput = (golput / (suara_1 + suara_2 + suara_3 + golput)
) * 100

  estimation_parameter_data.append({
    "Sample (%)": int(i / 20 * 100),
    "suara_1": percentage_suara_1,
    "suara_2": percentage_suara_2,
    "suara_3": percentage_suara_3,
    "golput": percentage_golput,
  })

# make estimation parameter data to dataframe
estimation_parameter_data = pd.DataFrame(estimation_parameter_data)

print("================= Estimasi Proporsi Parameter ================
==")
estimation_parameter_data
```

```
================= Estimasi Proporsi Parameter =================

    Sample (%)    suara_1    suara_2    suara_3    golput
0            5  44.462478  31.347090  23.776953  0.413479
1           10  43.188628  32.787100  23.591655  0.432617
2           15  42.110577  35.884679  21.564382  0.440363
3           20  40.945883  36.852451  21.738167  0.463498
4           25  41.320227  36.088118  22.143034  0.448621
..         ...        ...        ...        ...       ...
14          75  38.963209  38.229118  22.392039  0.415634
15          80  39.010659  38.258150  22.319643  0.411548
```

```
16          85  39.078069  38.098455  22.412804  0.410672
17          90  39.072301  38.007035  22.511599  0.409065
18          95  39.013837  38.179924  22.397068  0.409171

[19 rows x 5 columns]
```

Setelah berhasil mengestimasi parameter menggunakan metode ***2 stage random sampling***, hasilnya divisualisasikan dalam bentuk grafik estimasi parameter. Dengan grafik ini, dapat dilihat proporsi suara untuk setiap kandidat dan jumlah golput berubah seiring dengan peningkatan persentase sampel yang diambil.

```
# grafik estimasi parameter terhadap sample size
import matplotlib.pyplot as plt
import seaborn as sns

x = estimation_parameter_data["Sample (%)"]

plt.figure(figsize = (10, 5))
plt.plot(x, estimation_parameter_data["suara_1"], label = "Suara 1", c
olor = "green", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["suara_2"], label = "Suara 2", c
olor = "blue", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["suara_3"], label = "Suara 3", c
olor = "yellow", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["golput"], label = "Golput", col
or = "red", linestyle = "dashed")
plt.title("Grafik Estimasi Proporsi Parameter Terhadap Sample Size", s
ize = 15)
plt.xlabel("Sample (%)")
plt.ylabel("Error (%)")
plt.legend()
plt.show()
```



Berdasarkan plot hasil estimasi parameter, diketahui bahwa proporsi suara kandidat 1 cenderung menurun seiring dengan peningkatan ukuran sampel. Hal ini dapat

diindikasikan oleh penurunan persentase suara_1 dari sekitar 44.46% pada sampel 5% menjadi sekitar 39.01% pada sampel 80%. Meskipun ada fluktuasi, penurunan ini menggambarkan kecenderungan penurunan dukungan terhadap Kandidat 1 seiring dengan peningkatan ukuran sampel. Sebaliknya, proporsi suara kandidat 2 cenderung meningkat seiring dengan peningkatan ukuran sampel. Hal ini dapat dilihat dari peningkatan persentase suara_2 dari sekitar 31.35% pada sampel 5% menjadi sekitar 38.26% pada sampel 80%. Ini mengindikasikan kecenderungan peningkatan dukungan terhadap kandidat 2 seiring dengan peningkatan ukuran sampel. Proporsi suara kandidat 3 juga mengalami fluktuasi, tetapi tidak sekuat suara_1 dan suara_2. Hal ini mengindikasikan bahwa dukungan terhadap kandidat 3 relatif stabil dan kurang dipengaruhi oleh ukuran sampel. Persentase golput cenderung stabil dan rendah, berkisar antara 0.4% hingga 0.46%, meskipun ada fluktuasi kecil. Ini menunjukkan bahwa tingkat partisipasi pemilih dalam pemilihan relatif tinggi dan tidak banyak dipengaruhi oleh ukuran sampel.

Selanjutnya, melakukan perbandingan antara jumlah suara di kabupaten dengan jumlah suara di kecamatan dengan program berikut.

```
kab = (completed_data_kab["suara_1"].sum() + completed_data_kab["suara
_2"].sum() + completed_data_kab["suara_3"].sum() + completed_data_kab[
"golput"].sum())
kec = (completed_data_kec["suara_1"].sum() + completed_data_kec["suara
_2"].sum() + completed_data_kec["suara_3"].sum() + completed_data_kec[
"golput"].sum())

perbandingan = kab / kec
print(perbandingan)
```

```
1.912865799373941
```

Hasil perbandingan di atas mengindikasikan seberapa besar perbedaan antara jumlah total suara di tingkat kabupaten dengan jumlah total suara di tingkat kecamatan. Angka perbandingan sekitar 1.91 menunjukkan bahwa jumlah total suara di tingkat kabupaten sekitar 1.91 kali lebih besar daripada jumlah total suara di tingkat kecamatan. Dengan kata lain, secara keseluruhan, tingkat partisipasi pemilih atau jumlah total suara di tingkat kabupaten jauh lebih tinggi daripada di tingkat kecamatan.

Selanjutnya, menghitung *error* estimasi parameter guna mengukur tingkat ketidakpastian dan mengevaluasi metode *sampling*.

```
# hitung eror estimasi parameter suara_1, suara_2, suara_3, dan golput
percentage_suara_1_kabupaten = (completed_data_kec["suara_1"].sum() *
perbandingan / (completed_data_kab["suara_1"].sum() + completed_data_k
ab["suara_2"].sum() + completed_data_kab["suara_3"].sum() + completed_
data_kab["golput"].sum())) * 100
```

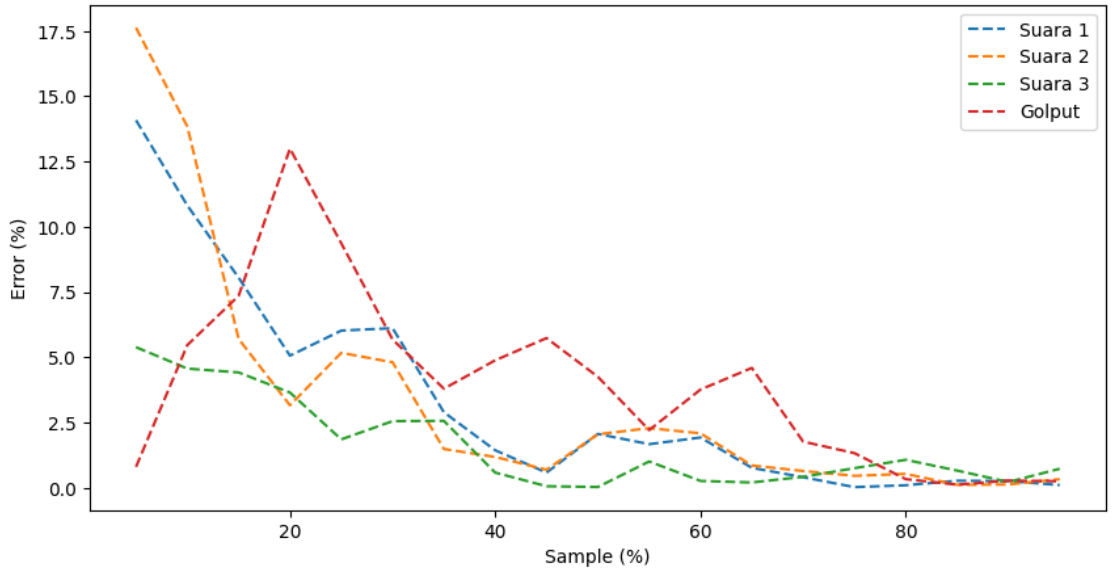
```
percentage_suara_2_kabupaten = (completed_data_kec["suara_2"].sum() *
perbandingan  / (completed_data_kab["suara_1"].sum() + completed_data_
kab["suara_2"].sum() + completed_data_kab["suara_3"].sum() + completed
_data_kab["golput"].sum())) * 100
percentage_suara_3_kabupaten = (completed_data_kec["suara_3"].sum() *
perbandingan / (completed_data_kab["suara_1"].sum() + completed_data_k
ab["suara_2"].sum() + completed_data_kab["suara_3"].sum() + completed_
data_kab["golput"].sum())) * 100
percentage_golput_kabupaten = (completed_data_kec["golput"].sum() * pe
rbandingan / (completed_data_kab["suara_1"].sum() + completed_data_kab
["suara_2"].sum() + completed_data_kab["suara_3"].sum() + completed_da
ta_kab["golput"].sum())) * 100

x = estimation_parameter_data["Sample (%)"]
error_suara_1 = abs(100 * (estimation_parameter_data["suara_1"] - perc
entage_suara_1_kabupaten) / percentage_suara_1_kabupaten)
error_suara_2 = abs(100 * (estimation_parameter_data["suara_2"] - perc
entage_suara_2_kabupaten) / percentage_suara_2_kabupaten)
error_suara_3 = abs(100 * (estimation_parameter_data["suara_3"] - perc
entage_suara_3_kabupaten) / percentage_suara_3_kabupaten)
error_golput = abs(100 * (estimation_parameter_data["golput"] - percen
tage_golput_kabupaten) / percentage_golput_kabupaten)

data_error = pd.DataFrame({
  "Sample (%)": x,
  "Error Suara 1": error_suara_1,
  "Error Suara 2": error_suara_2,
  "Error Suara 3": error_suara_3,
  "Error Golput": error_golput
})
data_error.to_csv('output_2MRS.csv', index = False)

print("============== Error Estimasi Proporsi Parameter ==============
")
data_error
```

```
============== Error Estimasi Proporsi Parameter ==============

    Sample (%)  Error Suara 1  Error Suara 2  Error Suara 3  Error Golput
0            5      14.084957      17.627514       5.387825      0.800976
1           10      10.816425      13.843520       4.566517      5.466595
2           15       8.050285       5.703842       4.419071      7.354933
3           20       5.061832       3.160772       3.648794     12.995154
4           25       6.022350       5.169253       1.854282      9.368235
..         ...            ...            ...            ...           ...
14          75       0.025454       0.456772       0.750604      1.326394
15          80       0.096296       0.533061       1.071488      0.330351
16          85       0.269262       0.113422       0.658567      0.116721
17          90       0.254463       0.126808       0.220672      0.275081
18          95       0.104451       0.327501       0.728311      0.249173

[19 rows x 5 columns]
```

```
# display estimation persentase error of proportion
fig = plt.figure(figsize = (10, 5))

x = estimation_parameter_data["Sample (%)"]

estimation_parameter_data["error_suara_1"] = error_suara_1
estimation_parameter_data["error_suara_2"] = error_suara_2
estimation_parameter_data["error_suara_3"] = error_suara_3
estimation_parameter_data["error_golput"] = error_golput
```

```
# graph error
plt.figure(figsize = (10, 5))
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_suara_1"], label = "Suara 1", linestyle = "dashed")
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_suara_2"], label = "Suara 2", linestyle = "dashed")
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_suara_3"], label = "Suara 3", linestyle = "dashed")
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_golput"], label = "Golput", linestyle = "dashed")
plt.xlabel("Sample (%)")
plt.ylabel("Error (%)")
plt.legend()
plt.show()
```

```
<Figure size 1000x500 with 0 Axes>
```



Berdasarkan plot di atas, estimasi *error* parameter menunjukkan tren menurunnya *error* estimasi seiring dengan peningkatan persentase sampel yang diambil. *Error* tertinggi terjadi pada 5% sampel dan secara konsisten mengalami penurunan hingga mencapai tingkat yang sangat rendah saat mengambil 75% sampel, sebelum sedikit naik pada 80% dan 85% sampel. Ini mengindikasikan bahwa semakin besar sampel yang diambil, semakin akurat estimasi parameter suara dan golputnya. Meskipun *error* pada 5% sampel pun relatif rendah, menunjukkan bahwa pengambilan sampel dalam *quick count* telah memberikan hasil yang kredibel dan dapat diandalkan dalam memantau hasil Pilkada di tingkat kabupaten.

Langkah selanjutnya melakukan pemisahan data berdasarkan kelurahan di setiap kecamatan dengan kode program sebagai berikut.

```
kel_adm_mliriprowo = kec_adm_tarik[kec_adm_tarik["kelurahan"] == "MLIR
IPROWO"]
kel_adm_kedungbocok = kec_adm_tarik[kec_adm_tarik["kelurahan"] == "KED
```

```
UNGBOCOK"]
kel_adm_singogalih = kec_adm_tarik[kec_adm_tarik["kelurahan"] == "SING
OGALIH"]
kel_adm_margosari = kec_adm_tarik[kec_adm_tarik["kelurahan"] == "MERGO
SARI"]
kel_adm_prambon = kec_adm_prambon[kec_adm_prambon["kelurahan"] == "PRA
MBON"]
kel_adm_kajartengguli = kec_adm_prambon[kec_adm_prambon["kelurahan"] =
= "KAJARTENGGULI"]
kel_adm_gedangrowo = kec_adm_prambon[kec_adm_prambon["kelurahan"] == "
GEDANGROWO"]
kel_adm_karangtanjung = kec_adm_candi[kec_adm_candi["kelurahan"] == "K
ARANGTANJUNG"]
kel_adm_sidokare = kec_adm_sidoarjo[kec_adm_sidoarjo["kelurahan"] == "
SIDOKARE"]
kel_adm_celep = kec_adm_sidoarjo[kec_adm_sidoarjo["kelurahan"] == "CEL
EP"]
kel_adm_kalisampurno = kec_adm_tanggulangin[kec_adm_tanggulangin["kelu
rahan"] == "KALISAMPURNO"]
kel_adm_kalitengah = kec_adm_tanggulangin[kec_adm_tanggulangin["kelura
han"] == "KALITENGAH"]
kel_adm_tanggul = kec_adm_wonoayu[kec_adm_wonoayu["kelurahan"] == "TAN
GGUL"]
kel_adm_pilang = kec_adm_wonoayu[kec_adm_wonoayu["kelurahan"] == "PILA
NG"]
kel_adm_wonokasian = kec_adm_wonoayu[kec_adm_wonoayu["kelurahan"] == "
WONOKASIAN"]
```

```
# Data TPS beserta suara di Kec Mliriprowo
suara_1 = 1022
suara_2 = 299
suara_3 = 761
golput = 12
data_suara_mliriprowo = generate_tps(kel_adm_mliriprowo, suara_1, suar
a_2, suara_3, golput)
data_suara_mliriprowo
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	MLIRIPROWO	1	43	188	183	2
1	MLIRIPROWO	2	238	27	277	2
2	MLIRIPROWO	3	338	56	243	3
3	MLIRIPROWO	4	403	28	59	5
4	TOTAL	4	1022	299	762	12

```
# Data TPS beserta suara di Kec kedungbocok
suara_1 = 626
suara_2 = 484
suara_3 = 939
golput = 10
data_suara_kedungbocok = generate_tps(kel_adm_kedungbocok, suara_1, su
ara_2, suara_3, golput)
data_suara_kedungbocok
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	KEDUNGBOCOK	1	178	25	78	2
1	KEDUNGBOCOK	2	128	210	517	2
2	KEDUNGBOCOK	3	28	89	61	1
3	KEDUNGBOCOK	4	146	134	83	1

4	KEDUNGBOCOK	5	146	26	200	4
5	TOTAL	5	626	484	939	10

```
# Data TPS beserta suara di Kec Singogalih
suara_1 = 1007
suara_2 = 1202
suara_3 = 587
golput = 9
data_suara_singogalih = generate_tps(kel_adm_singogalih, suara_1, suar
a_2, suara_3, golput)
data_suara_singogalih
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	SINGOGALIH	1	100	82	393	0
1	SINGOGALIH	2	316	549	7	5
2	SINGOGALIH	3	139	255	17	2
3	SINGOGALIH	4	452	316	169	2
4	TOTAL	4	1007	1202	586	9

```
# Data TPS beserta suara di Kel Margosari
suara_1 = 956
suara_2 = 583
suara_3 = 457
golput = 13
data_suara_margosari = generate_tps(kel_adm_margosari, suara_1, suara_
2, suara_3, golput)
data_suara_margosari
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	MERGOSARI	1	236	22	93	2
1	MERGOSARI	2	57	78	59	1
2	MERGOSARI	3	119	32	50	0
3	MERGOSARI	4	5	34	73	7
4	MERGOSARI	5	202	56	74	1
5	MERGOSARI	6	85	201	19	1
6	MERGOSARI	7	252	159	89	1
7	TOTAL	7	956	582	457	13

```
# Data TPS beserta suara di Kel Prambon
suara_1 = 305
suara_2 = 1256
suara_3 = 623
golput = 14
data_suara_prambon = generate_tps(kel_adm_prambon, suara_1, suara_2, s
uara_3, golput)
data_suara_prambon
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	PRAMBON	1	65	468	197	2
1	PRAMBON	2	6	375	213	4
2	PRAMBON	3	22	217	98	3
3	PRAMBON	4	146	58	20	3
4	PRAMBON	5	16	99	21	0
5	PRAMBON	6	50	40	73	3
6	TOTAL	6	305	1257	622	15

```
# Data TPS beserta suara di Kel Kajartengguli
suara_1 = 726
suara_2 = 579
suara_3 = 294
golput = 18
data_suara_kajartengguli = generate_tps(kel_adm_kajartengguli, suara_1
, suara_2, suara_3, golput)
data_suara_kajartengguli
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	KAJARTENGGULI	1	117	55	6	0
1	KAJARTENGGULI	2	65	236	26	0
2	KAJARTENGGULI	3	9	86	80	7
3	KAJARTENGGULI	4	190	44	93	0
4	KAJARTENGGULI	5	125	34	56	3
5	KAJARTENGGULI	6	134	5	8	6
6	KAJARTENGGULI	7	85	119	24	1
7	TOTAL	7	725	579	293	17

```
# Data TPS beserta suara di Kel Gedangworo
suara_1 = 539
suara_2 = 704
suara_3 = 643
golput = 12
data_suara_gedangrowo = generate_tps(kel_adm_gedangrowo, suara_1, suar
a_2, suara_3, golput)
data_suara_gedangrowo
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	GEDANGROWO	1	13	40	230	2
1	GEDANGROWO	2	200	206	111	0
2	GEDANGROWO	3	107	52	18	4
3	GEDANGROWO	4	49	313	53	0
4	GEDANGROWO	5	171	93	231	5
5	TOTAL	5	540	704	643	11

```
# Data TPS beserta suara di Kel Karangtanjung
suara_1 = 1043
suara_2 = 1821
suara_3 = 708
golput = 17
data_suara_karangtanjung = generate_tps(kel_adm_karangtanjung, suara_1
, suara_2, suara_3, golput)
data_suara_karangtanjung
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	KARANGTANJUNG	1	40	88	134	3
1	KARANGTANJUNG	2	31	1247	214	2
2	KARANGTANJUNG	3	458	58	155	4
3	KARANGTANJUNG	4	106	324	3	5
4	KARANGTANJUNG	5	408	104	201	3
5	TOTAL	5	1043	1821	707	17

```
# Data TPS beserta suara di Kel Sidokare
suara_1 = 2580
suara_2 = 1702
suara_3 = 1659
golput = 19
data_suara_sidokare = generate_tps(kel_adm_sidokare, suara_1, suara_2,
suara_3, golput)
data_suara_sidokare
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	SIDOKARE	1	377	216	129	6
1	SIDOKARE	2	166	236	1278	1
2	SIDOKARE	3	1360	400	236	6
3	SIDOKARE	4	677	850	16	5
4	TOTAL	4	2580	1702	1659	18

```
# Data TPS beserta suara di Kel Celep
suara_1 = 1252
suara_2 = 1084
suara_3 = 429
golput = 14
data_suara_celep = generate_tps(kel_adm_celep, suara_1, suara_2, suara
_3, golput)
data_suara_celep
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	CELEP	1	2	316	59	3
1	CELEP	2	433	98	263	0
2	CELEP	3	44	223	82	8
3	CELEP	4	255	199	20	2
4	CELEP	5	518	248	5	0
5	TOTAL	5	1252	1084	429	13

```
# Data TPS beserta suara di Kel Kalisampurno
suara_1 = 1192
suara_2 = 2162
suara_3 = 568
golput = 18
data_suara_kalisampurno = generate_tps(kel_adm_kalisampurno, suara_1,
suara_2, suara_3, golput)
data_suara_kalisampurno
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	KALISAMPURNO	1	251	234	77	1
1	KALISAMPURNO	2	115	229	25	1
2	KALISAMPURNO	3	141	2	47	5
3	KALISAMPURNO	4	387	1001	147	0
4	KALISAMPURNO	5	82	592	244	1
5	KALISAMPURNO	6	43	45	15	3
6	KALISAMPURNO	7	172	58	12	8
7	TOTAL	7	1191	2161	567	19

```
# Data TPS beserta suara di Kel Kalitenggah
suara_1 = 3119
suara_2 = 1248
suara_3 = 1518
```

```
golput = 20
data_suara_kalitengah = generate_tps(kel_adm_kalitengah, suara_1, suar
a_2, suara_3, golput)
data_suara_kalitengah
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	KALITENGAH	1	434	245	340	4
1	KALITENGAH	2	453	172	311	12
2	KALITENGAH	3	446	157	359	1
3	KALITENGAH	4	1479	385	112	0
4	KALITENGAH	5	306	289	396	3
5	TOTAL	5	3118	1248	1518	20

```
# Data TPS beserta suara di Kel Tanggul
suara_1 = 1051
suara_2 = 816
suara_3 = 582
golput = 16
data_suara_tanggul = generate_tps(kel_adm_tanggul, suara_1, suara_2, s
uara_3, golput)
data_suara_tanggul
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	TANGGUL	1	10	71	196	5
1	TANGGUL	2	119	360	141	3
2	TANGGUL	3	213	124	62	6
3	TANGGUL	4	709	262	183	2
4	TOTAL	4	1051	817	582	16

```
# Data TPS beserta suara di Kel Pilang
suara_1 = 1280
suara_2 = 1191
suara_3 = 473
golput = 13
data_suara_pilang = generate_tps(kel_adm_pilang, suara_1, suara_2, sua
ra_3, golput)
data_suara_pilang
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	PILANG	1	201	340	111	0
1	PILANG	2	763	45	107	0
2	PILANG	3	38	419	127	5
3	PILANG	4	90	228	6	3
4	PILANG	5	189	159	123	4
5	TOTAL	5	1281	1191	474	12

```
# Data TPS beserta suara di Kel wonokasian
suara_1 = 1663
suara_2 = 1109
suara_3 = 503
golput = 15
data_suara_wonokasian = generate_tps(kel_adm_wonokasian, suara_1, suar
a_2, suara_3, golput)
data_suara_wonokasian
```

	kelurahan	no_tps	suara_1	suara_2	suara_3	golput
0	WONOKASIAN	1	636	280	15	11
1	WONOKASIAN	2	59	38	2	2
2	WONOKASIAN	3	887	456	438	1
3	WONOKASIAN	4	82	335	47	1
4	TOTAL	4	1664	1109	502	15

Data suara yang diperoleh dari berbagai kelurahan di beberapa kecamatan menunjukkan variasi yang signifikan dalam perolehan suara. Dapat diketahui bahwa beberapa kelurahan memiliki jumlah suara yang signifikan untuk kandidat tertentu,seperti kelurahan Singogalih yang mendukung suara_1 dengan suara terbanyak, sementara kelurahan Kajartengguli dan Kalitengah memiliki persentase golput yang rendah. Di sisi lain, beberapa kelurahan lainnya memiliki variasi suara yang lebih merata antara setiap kandidat, seperti di kelurahan Prambon dan Gedangrowo.

Setelahnya, data suara yang didapat digabungkan menjadi satu kesatuan dalam sebuah *dataframe*. Selain itu, baris "**TOTAL**" juga dihapus agar terfokus pada data suara per TPS yang relevan.

```
# collect all data suara and remove total row from every data
data_suara = pd.concat([data_suara_mliriprowo, data_suara_margosari, d
ata_suara_singogalih, data_suara_kedungbocok,
                        data_suara_prambon, data_suara_kajartengguli,
data_suara_gedangrowo, data_suara_karangtanjung,
                        data_suara_sidokare, data_suara_celep, data_su
ara_kalisampurno, data_suara_kalitengah,
                        data_suara_tanggul, data_suara_pilang, data_su
ara_wonokasian])
data_suara = data_suara[data_suara["kelurahan"] != "TOTAL"]

total_row = pd.DataFrame([{
  "kabupaten": "TOTAL",
  "kecamatan": len(data["kecamatan"].unique()),
  "kelurahan": len(data_suara["kelurahan"].unique()),
  "no_tps": len(data_suara["kelurahan"].unique()) * len(data_suara["no
_tps"].unique()),
  "suara_1": data_suara["suara_1"].sum(),
  "suara_2": data_suara["suara_2"].sum(),
  "suara_3": data_suara["suara_3"].sum(),
  "golput": data_suara["golput"].sum()
}])

# concat data suara with data
completed_data_kel = pd.merge(data, data_suara, on = "kelurahan", how
= "outer")
completed_data_kel = pd.concat([completed_data_kel, total_row], ignore
_index = True)
```

Pada langkah ini, estimasi proporsi parameter suara pemilihan dilakukan dengan menggunakan teknik pengambilan sampel acak. Kode menghasilkan data estimasi untuk berbagai ukuran sampel, mulai dari 5% hingga 95% dari data asli. Estimasi tersebut

mencakup proporsi suara untuk setiap kandidat (suara_1, suara_2, dan suara_3) serta persentase golput.

```
# estimasi parameter
estimation_parameter_data = []

for i in range(1, 20):
  random_sampling = completed_data_kec.sample(frac = i / 20, random_st
ate = 42).reset_index(drop = True)

  # sum all suara_1, suara_2, suara3, and golput
  suara_1 = random_sampling["suara_1"].sum()
  suara_2 = random_sampling["suara_2"].sum()
  suara_3 = random_sampling["suara_3"].sum()
  golput = random_sampling["golput"].sum()

  # calculate percentage of suara_1, suara_2, suara 3, and golput
  percentage_suara_1 = (suara_1 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_suara_2 = (suara_2 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_suara_3 = (suara_3 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_golput = (golput / (suara_1 + suara_2 + suara_3 + golput)
) * 100

  estimation_parameter_data.append({
    "Sample (%)": int(i / 20 * 100),
    "suara_1": percentage_suara_1,
    "suara_2": percentage_suara_2,
    "suara_3": percentage_suara_3,
    "golput": percentage_golput,
  })

# make estimation parameter data to dataframe
estimation_parameter_data = pd.DataFrame(estimation_parameter_data)

print("================== Estimasi Proporsi Parameter ================
==")
estimation_parameter_data
```

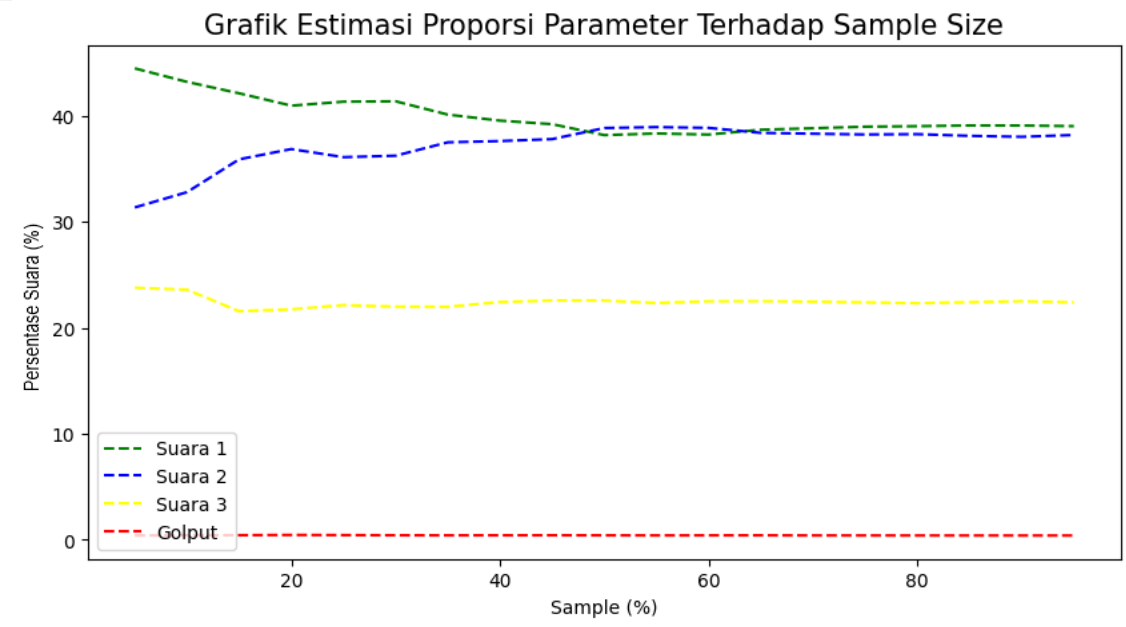
```
================== Estimasi Proporsi Parameter ==================

    Sample (%)    suara_1    suara_2    suara_3    golput
0            5  44.462478  31.347090  23.776953  0.413479
1           10  43.188628  32.787100  23.591655  0.432617
2           15  42.110577  35.884679  21.564382  0.440363
3           20  40.945883  36.852451  21.738167  0.463498
4           25  41.320227  36.088118  22.143034  0.448621
..         ...        ...        ...        ...       ...
14          75  38.963209  38.229118  22.392039  0.415634
15          80  39.010659  38.258150  22.319643  0.411548
16          85  39.078069  38.098455  22.412804  0.410672
17          90  39.072301  38.007035  22.511599  0.409065
18          95  39.013837  38.179924  22.397068  0.409171

[19 rows x 5 columns]
```

Hasil estimasi menunjukkan bahwa dengan meningkatnya ukuran sampel, proporsi suara untuk setiap kandidat cenderung lebih stabil dan mendekati nilai sesungguhnya.

Sebagai contoh, saat mengambil 5% sampel, proporsi suara mungkin lebih bervariasi, tetapi saat sampel mencapai 95%, proporsinya lebih mendekati hasil sebenarnya.

Setelah berhasil mengestimasi parameter menggunakan metode **3 *stage random sampling***, hasilnya divisualisasikan dalam bentuk grafik estimasi parameter. Dalam grafik ini, dapat dilihat bagaimana proporsi suara untuk setiap kandidat dan jumlah golput berubah seiring dengan peningkatan persentase sampel yang diambil.

```
# grafik estimasi parameter terhadap sample size
import matplotlib.pyplot as plt
import seaborn as sns

x = estimation_parameter_data["Sample (%)"]

plt.figure(figsize = (10, 5))
plt.plot(x, estimation_parameter_data["suara_1"], label = "Suara 1", c
olor = "green", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["suara_2"], label = "Suara 2", c
olor = "blue", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["suara_3"], label = "Suara 3", c
olor = "yellow", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["golput"], label = "Golput", col
or = "red", linestyle = "dashed")
plt.title("Grafik Estimasi Proporsi Parameter Terhadap Sample Size", s
ize = 15)
plt.xlabel("Sample (%)")
plt.ylabel("Persentase Suara (%)")
plt.legend()
plt.show()
```



Berdasarkan grafik estimasi proporsi di atas, proporsi suara_1 cenderung stabil seiring dengan peningkatan ukuran sampel. Pada awalnya, saat hanya mengambil 5% sampel, proporsi suara_1 bisa bervariasi, tetapi seiring dengan peningkatan sampel, grafiknya menjadi lebih halus dan mendekati nilai yang lebih konsisten. Seperti suara_1,

proporsi suara_2 juga mengalami peningkatan stabilitas dengan ukuran sampel yang lebih besar. Variabilitasnya menurun seiring dengan meningkatnya persentase sampel. Proporsi suara_3 juga menunjukkan tren yang serupa. Semakin besar sampel yang diambil, semakin stabil proporsi suara_3 yang diestimasi. Persentase golput cenderung stabil dan rendah, meskipun ada fluktuasi kecil. Ini menunjukkan bahwa tingkat partisipasi pemilih dalam pemilihan relatif tinggi dan tidak banyak dipengaruhi oleh ukuran sampel.

Selanjutnya, melakukan perbandingan antara jumlah suara di kabupaten dengan jumlah suara di kelurahan dengan program berikut.

```
kel = (completed_data_kel["suara_1"].sum() + completed_data_kel["suara
_2"].sum() + completed_data_kel["suara_3"].sum() + completed_data_kel[
"golput"].sum())

perbandingan2 = kab / kel
print(perbandingan2)

21.93069274331621
```

Hasil perbandingan di atas mengindikasikan seberapa besar perbedaan antara jumlah total suara di tingkat kabupaten dengan jumlah total suara di tingkat kelurahan. Angka perbandingan sekitar 21.9 menunjukkan bahwa jumlah total suara di tingkat kabupaten sekitar 21.9 kali lebih besar daripada jumlah total suara di tingkat kelurahan. Dengan kata lain, secara keseluruhan, tingkat partisipasi pemilih atau jumlah total suara di tingkat kabupaten jauh lebih tinggi daripada di tingkat kelurahan.

Selanjutnya, menghitung *error* estimasi parameter guna mengukur tingkat ketidakpastian dan mengevaluasi metode *sampling*.

```
# hitung eror estimasi parameter suara_1, suara_2, suara 3, dan golput
percentage_suara_1_kabupaten = (completed_data_kel["suara_1"].sum() *
perbandingan2 / (completed_data_kab["suara_1"].sum() + completed_data_
kab["suara_2"].sum() + completed_data_kab["suara_3"].sum() + completed
_data_kab["golput"].sum())) * 100
percentage_suara_2_kabupaten = (completed_data_kel["suara_2"].sum() *
perbandingan2  / (completed_data_kab["suara_1"].sum() + completed_data
_kab["suara_2"].sum() + completed_data_kab["suara_3"].sum() + complete
d_data_kab["golput"].sum())) * 100
percentage_suara_3_kabupaten = (completed_data_kel["suara_3"].sum() *
perbandingan2 / (completed_data_kab["suara_1"].sum() + completed_data_
kab["suara_2"].sum() + completed_data_kab["suara_3"].sum() + completed
_data_kab["golput"].sum())) * 100
percentage_golput_kabupaten = (completed_data_kel["golput"].sum() * pe
rbandingan2 / (completed_data_kab["suara_1"].sum() + completed_data_ka
b["suara_2"].sum() + completed_data_kab["suara_3"].sum() + completed_d
ata_kab["golput"].sum())) * 100

x = estimation_parameter_data["Sample (%)"]
error_suara_1 = abs(100 * (estimation_parameter_data["suara_1"] - perc
```

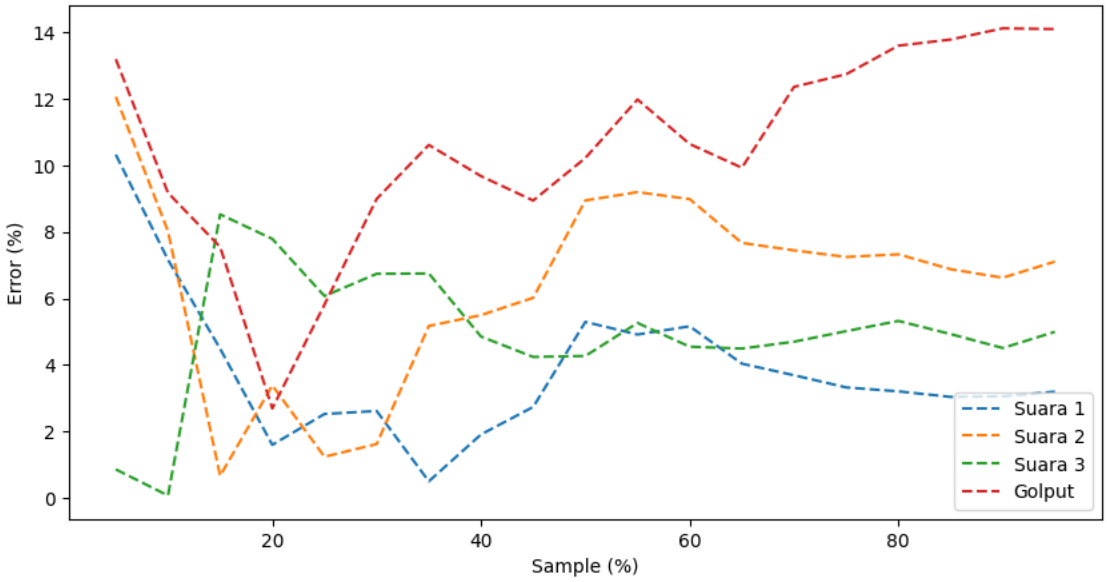
```
entage_suara_1_kabupaten) / percentage_suara_1_kabupaten)
error_suara_2 = abs(100 * (estimation_parameter_data["suara_2"] - perc
entage_suara_2_kabupaten) / percentage_suara_2_kabupaten)
error_suara_3 = abs(100 * (estimation_parameter_data["suara_3"] - perc
entage_suara_3_kabupaten) / percentage_suara_3_kabupaten)
error_golput = abs(100 * (estimation_parameter_data["golput"] - percen
tage_golput_kabupaten) / percentage_golput_kabupaten)

data_error = pd.DataFrame({
  "Sample (%)": x,
  "Error Suara 1": error_suara_1,
  "Error Suara 2": error_suara_2,
  "Error Suara 3": error_suara_3,
  "Error Golput": error_golput
})

data_error.to_csv('output_3MRS.csv', index = False)

print("============== Error Estimasi Proporsi Parameter ==============
")
data_error
```

```
============== Error Estimasi Proporsi Parameter ==============

    Sample (%)  Error Suara 1  Error Suara 2  Error Suara 3  Error Golput
0            5      10.321963      12.062147       0.859445     13.192353
1           10       7.161240       8.022493       0.073427      9.174422
2           15       4.486339       0.667130       8.526061      7.548225
3           20       1.596458       3.382019       7.788882      2.690987
4           25       2.525294       1.237838       6.071478      5.814412
..         ...            ...            ...            ...           ...
14          75       3.323028       7.243976       5.015223     12.739874
15          80       3.205294       7.325419       5.322319     13.597645
16          85       3.038033       6.877428       4.927141     13.781618
17          90       3.052344       6.620967       4.508062     14.119030
18          95       3.197408       7.105970       4.993889     14.096718

[19 rows x 5 columns]
```

```
# display estimation persentase error of proportion
fig = plt.figure(figsize = (10, 5))

x = estimation_parameter_data["Sample (%)"]

estimation_parameter_data["error_suara_1"] = error_suara_1
estimation_parameter_data["error_suara_2"] = error_suara_2
estimation_parameter_data["error_suara_3"] = error_suara_3
estimation_parameter_data["error_golput"] = error_golput

# graph error
plt.figure(figsize = (10, 5))
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_suara_1"], label = "Suara 1", linestyle = "dashed")
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_suara_2"], label = "Suara 2", linestyle = "dashed")
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_suara_3"], label = "Suara 3", linestyle = "dashed")
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_golput"], label = "Golput", linestyle = "dashed")
plt.xlabel("Sample (%)")
plt.ylabel("Error (%)")
```

```
plt.legend()
plt.show()
```

```
<Figure size 1000x500 with 0 Axes>
```



Berdasarkan grafik di atas, estimasi *error* parameter menunjukkan fluktuasi yang cenderung tidak stabil. *Error* terendah terdapat saat menggunakan sampel sekitar 20%, setelah melewati persentase itu, seiring bertambahnya sampel, *error* mengalami fluktuasi yang cenderung naik, terutama pada suara golput.

C. Menaksir Estimasi Parameter Menggunakan Metode *Cluster Random Sampling*

Pada tahap ini, estimasi parameter dilakukan menggunakan metode *cluster random sampling*, yaitu dengan mengelompokkan *dataset* berdasarkan kabupaten, kecamatan, dan kelurahan.

```
# make cluster sampling from completed_data
cluster_sampling = completed_data.groupby(["provinsi", "kabupaten", "k
ecamatan"]).apply(lambda x: x.sample(frac = .1, replace = False))
cluster_sampling = cluster_sampling.reset_index(drop = True)
cluster_sampling
```

```
                provinsi kabupaten    kecamatan         kelurahan  no_tps
\
0    PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO         WATESARI       1
1    PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO   SUMOKEMBANGSRI       3
2    PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO       WONOKUPANG       4
3    PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO           SEKETI       2
4    PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO        KEMANGSEN       2
..                   ...       ...          ...              ...     ...
190  PROVINSI JAWA TIMUR  SIDOARJO      WONOAYU          PLAOSAN       2
191  PROVINSI JAWA TIMUR  SIDOARJO      WONOAYU          TANGGUL       1
192  PROVINSI JAWA TIMUR  SIDOARJO      WONOAYU  SIMOANGIN-ANGIN       3
193  PROVINSI JAWA TIMUR  SIDOARJO      WONOAYU            PLOSO       2
194  PROVINSI JAWA TIMUR  SIDOARJO      WONOAYU       SUMBERREJO       4

     suara_1  suara_2  suara_3  golput
0         20      263       31       0
```

```
1          25     1131        45        1
2         699      174        59        4
3         229      713       128        0
4         271      199       204        1
..        ...      ...       ...      ...
190        25      158       192        4
191        25      215        39        1
192       238      597       132        1
193       636      199       214        1
194       196      280        48        2

[195 rows x 9 columns]
```

Selanjutnya, dilakukan estimasi parameter dari setiap kandidat (suara_1, suara_2, dan suara_3) serta persentase golput. Estimasi yang dihasilkan terdiri dari berbagai ukuran sampel, mulai dari 5% hingga 95% dari data asli.

```
# estimasi parameter
estimation_parameter_data = []
estimasi_suara_1 = []
estimasi_suara_2 = []
estimasi_suara_3 = []
estimasi_golput = []

for i in range(1, 20):
  cluster_sampling = completed_data.groupby(["provinsi", "kabupaten",
"kecamatan"]).apply(lambda x: x.sample(frac = i / 20, replace = False)
)

  # sum all suara_1, suara_2, suara 3, and golput
  suara_1 = cluster_sampling["suara_1"].sum()
  suara_2 = cluster_sampling["suara_2"].sum()
  suara_3 = cluster_sampling["suara_3"].sum()
  golput = cluster_sampling["golput"].sum()

  # append to list
  estimasi_suara_1.append(suara_1)
  estimasi_suara_2.append(suara_2)
  estimasi_suara_3.append(suara_3)
  estimasi_golput.append(golput)

  # calculate percentage of suara_1, suara_2, suara 3, and golput
  percentage_suara_1 = (suara_1 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_suara_2 = (suara_2 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_suara_3 = (suara_3 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_golput = (golput / (suara_1 + suara_2 + suara_3 + golput)
) * 100

  estimation_parameter_data.append({
    "Sample (%)": int(i / 20 * 100),
    "suara_1": percentage_suara_1,
    "suara_2": percentage_suara_2,
    "suara_3": percentage_suara_3,
    "golput": percentage_golput,
  })

# make estimation parameter data to dataframe
```

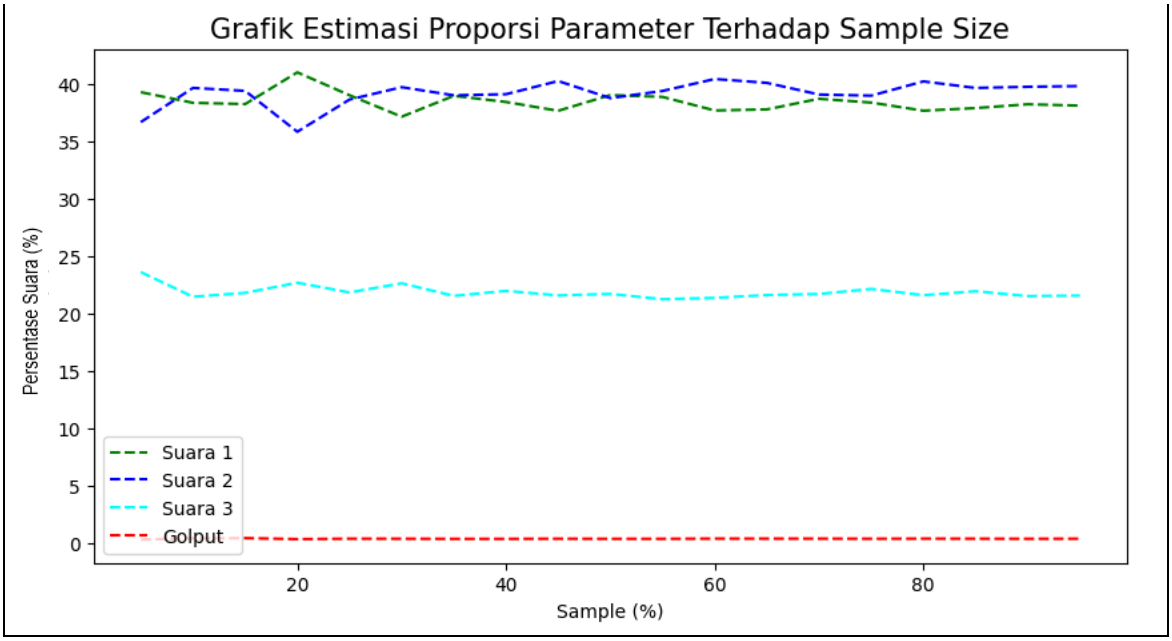
```
estimation_parameter_data = pd.DataFrame(estimation_parameter_data)

print("================= Estimasi Proporsi Parameter ================
==")
estimation_parameter_data
```

```
================= Estimasi Proporsi Parameter =================

    Sample (%)    suara_1    suara_2    suara_3    golput
0            5  39.304441  36.697486  23.640474  0.357599
1           10  38.378961  39.674354  21.494684  0.452001
2           15  38.267956  39.419127  21.832924  0.479993
3           20  41.029806  35.859115  22.717763  0.393315
4           25  39.040070  38.662215  21.868105  0.429610
..         ...        ...        ...        ...       ...
14          75  38.398717  39.002405  22.173931  0.424948
15          80  37.693567  40.242303  21.628793  0.435337
16          85  37.919525  39.674362  21.975981  0.430132
17          90  38.252367  39.774789  21.550723  0.422122
18          95  38.133646  39.838771  21.599433  0.428151

[19 rows x 5 columns]
```

Pada tingkat sampel 5%, proporsi suara untuk suara_1 berkisar antara 39% hingga 41%, suara_2 berkisar antara 35% hingga 40%, dan suara_3 berkisar antara 21% hingga 23%. Golput pada tingkat ini cenderung rendah, berkisar antara 0.35% hingga 0.45%. Seiring dengan peningkatan persentase sampel yang diambil, variasi dalam estimasi parameter sedikit berkurang, dan proporsi suara untuk masing-masing kandidat serta golput menjadi lebih stabil. Namun, tetap terdapat variasi dalam estimasi di setiap tingkat sampel. Secara keseluruhan, metode *cluster random sampling* memberikan estimasi parameter yang lebih detail dan representatif terhadap proporsi suara dan golput di berbagai kelurahan dan tingkat kabupaten dalam Pilkada.

Grafik mengenai hasil estimasi parameter adalah sebagai berikut.

```
# grafik estimasi parameter terhadap sample size
import matplotlib.pyplot as plt
import seaborn as sns

x = estimation_parameter_data["Sample (%)"]

plt.figure(figsize=(10, 5))
plt.plot(x, estimation_parameter_data["suara_1"], label = "Suara 1", c
olor = "green", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["suara_2"], label = "Suara 2", c
olor = "blue", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["suara_3"], label = "Suara 3", c
olor = "cyan", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["golput"], label = "Golput", col
or = "red", linestyle = "dashed")
plt.title("Grafik Estimasi Proporsi Parameter Terhadap Sample Size", s
ize = 15)
plt.xlabel("Sample (%)")
plt.ylabel("Persentase Suara (%)")
plt.legend()
plt.show()
```



Pada grafik di atas, dapat dilihat bahwa estimasi parameter suara_1 cenderung mengalami fluktuasi seiring dengan peningkatan persentase sampel yang diambil. Hal yang sama juga berlaku untuk suara_2 dan suara_3. Namun, suara_3 menunjukkan fluktuasi yang lebih signifikan pada tingkat sampel yang lebih rendah (5% hingga 25%) sebelum mulai stabil pada tingkat sampel yang lebih tinggi. Sedangkan untuk suara golput, cenderung stabil dan memiliki nilai yang rendah di setiap sampelnya.

```
# hitung eror estimasi parameter suara_1, suara_2, suara 3, dan golput
percentage_suara_1_kabupaten = (completed_data["suara_1"].sum() / (com
pleted_data["suara_1"].sum() + completed_data["suara_2"].sum()  + comp
leted_data["suara_3"].sum() + completed_data["golput"].sum())) * 100
percentage_suara_2_kabupaten = (completed_data["suara_2"].sum() / (com
pleted_data["suara_1"].sum() + completed_data["suara_2"].sum()  + comp
leted_data["suara_3"].sum() + completed_data["golput"].sum())) * 100
percentage_suara_3_kabupaten = (completed_data["suara_3"].sum() / (com
pleted_data["suara_1"].sum() + completed_data["suara_2"].sum()  + comp
leted_data["suara_3"].sum() + completed_data["golput"].sum())) * 100
percentage_golput_kabupaten = (completed_data["golput"].sum() / (compl
eted_data["suara_1"].sum() + completed_data["suara_2"].sum()  + comple
ted_data["suara_3"].sum() + completed_data["golput"].sum())) * 100

x = estimation_parameter_data["Sample (%)"]
error_suara_1 = abs(100 * (estimation_parameter_data["suara_1"] - perc
entage_suara_1_kabupaten) / percentage_suara_1_kabupaten)
error_suara_2 = abs(100 * (estimation_parameter_data["suara_2"] - perc
entage_suara_2_kabupaten) / percentage_suara_2_kabupaten)
error_suara_3 = abs(100 * (estimation_parameter_data["suara_3"] - perc
entage_suara_3_kabupaten) / percentage_suara_3_kabupaten)
error_golput = abs(100 * (estimation_parameter_data["golput"] - percen
tage_golput_kabupaten) / percentage_golput_kabupaten)

data_error = pd.DataFrame({
  "Sample (%)": x,
  "Error Suara 1": error_suara_1,
  "Error Suara 2": error_suara_2,
```

```
    "Error Suara 3": error_suara_3,
  "Error Golput": error_golput
})

data_error.to_csv('output_CRS.csv', index = False)

print("============== Error Estimasi Proporsi Parameter ==============
")
data_error
```

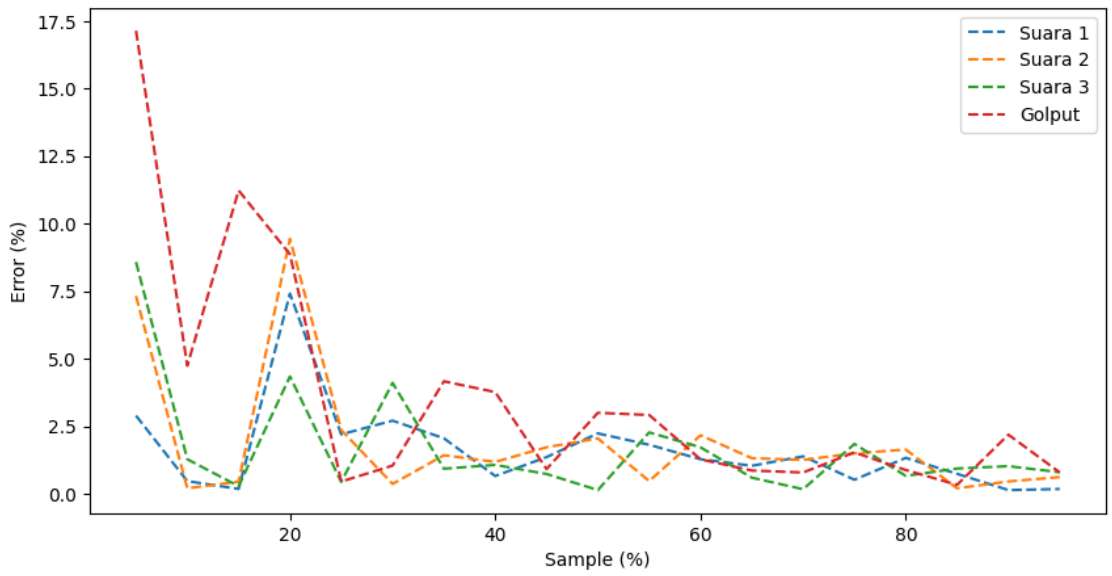
```
============== Error Estimasi Proporsi Parameter ==============

    Sample (%)  Error Suara 1  Error Suara 2  Error Suara 3  Error Golput
0            5       2.889350       7.318065       8.578619     17.136538
1           10       0.466672       0.200214       1.276805      4.738237
2           15       0.176086       0.444381       0.276704     11.224786
3           20       7.405931       9.435425       4.340691      8.860451
4           25       2.197291       2.356011       0.438285      0.450049
..         ...            ...            ...            ...           ...
14          75       0.518386       1.496840       1.842916      1.530376
15          80       1.327522       1.634607       0.660853      0.876840
16          85       0.736019       0.200233       0.933752      0.329239
17          90       0.135278       0.453867       1.019424      2.185264
18          95       0.175504       0.615460       0.795703      0.788332

[19 rows x 5 columns]
```

Pada awalnya, dengan hanya mengambil 5% sampel dari data, terdapat kesalahan estimasi yang cukup besar, khususnya pada suara kandidat pertama dan kedua, serta golput. Kesalahan ini secara signifikan lebih rendah saat sampel mencapai 10%, yang mengindikasikan bahwa pengambilan sampel lebih besar menghasilkan estimasi yang lebih akurat. Namun, pada tingkat sampel yang lebih tinggi, seperti 75% hingga 95%, tingkat kesalahan relatif kecil, yang berarti hasil estimasi sudah cukup mendekati nilai sebenarnya.

```
# display estimation persentase error of proportion
fig = plt.figure(figsize = (10, 5))

x = estimation_parameter_data["Sample (%)"]

estimation_parameter_data["error_suara_1"] = error_suara_1
estimation_parameter_data["error_suara_2"] = error_suara_2
estimation_parameter_data["error_suara_3"] = error_suara_3
estimation_parameter_data["error_golput"] = error_golput

# graph error
plt.figure(figsize = (10, 5))
plt.plot(x, estimation_parameter_data["error_suara_1"], label = "Suara
1", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["error_suara_2"], label = "Suara
2", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["error_suara_3"], label = "Suara
3", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["error_golput"], label = "Golput
", linestyle = "dashed")
plt.xlabel("Sample (%)")
plt.ylabel("Error (%)")
```

```
plt.legend()
plt.show()
```

```
<Figure size 1000x500 with 0 Axes>
```



Berdasarkan grafik di atas, dapat dilihat bahwa *error* suara_1, suara_2, dan suara_3 cenderung menurun seiring dengan peningkatan persentase sampel yang diambil. Pada tingkat sampel 5%, *error* untuk suara_1 adalah sekitar 2.89%, sedangkan pada tingkat sampel 95%, *error*-nya turun menjadi sekitar 0.18%, menunjukkan peningkatan signifikan dalam akurasi estimasi. *Error* golput, meskipun mengalami penurunan seiring dengan peningkatan persentase sampel, tetap memiliki tingkat fluktuasi yang relatif tinggi pada tingkat sampel yang lebih rendah (5% hingga 25%). Namun, *error* golput juga turun drastis seiring dengan peningkatan sampel dan menjadi kurang dari 1% pada tingkat sampel 95%. Hal ini mengindikasikan bahwa semakin besar sampel yang diambil, semakin akurat estimasi parameter suara dan golputnya. Meskipun error pada tingkat sampel yang lebih rendah relatif rendah, hasilnya tetap menunjukkan bahwa pengambilan sampel dalam *cluster random sampling* memberikan hasil yang kredibel dan dapat diandalkan dalam memantau hasil Pilkada di tingkat kabupaten.

D. Menaksir Estimasi Parameter Menggunakan Metode *Multistage Cluster Random Sampling*

Metode yang terakhir adalah metode *multistage cluster random sampling* yang dilakukan dengan kode sebagai berikut.

```
# get 3 random kecamatan from each kabupaten_kota as make it to list,
take all kecamatan in that city
completed_data = completed_data[completed_data["provinsi"] != "TOTAL"]

# get random kelurahan from completed_data as make it to list
```

```
kelurahan_list = np.random.choice(completed_data["kelurahan"].unique()
, 50, replace = False)

# take only 3 random tps from each kelurahan as dataframe from complet
ed_data
tps = []
for kel in kelurahan_list:
  tps.append(completed_data[completed_data["kelurahan"] == kel].sample
(n = 2, replace = False))

# concat all tps to one dataframe
cluster_sampling = pd.concat(tps).reset_index(drop = True)
cluster_sampling
```

```
               provinsi kabupaten    kecamatan        kelurahan  no_tps  \
0   PROVINSI JAWA TIMUR  SIDOARJO      BUDURAN  BANJARKEMANTREN       1
1   PROVINSI JAWA TIMUR  SIDOARJO      BUDURAN  BANJARKEMANTREN       4
2   PROVINSI JAWA TIMUR  SIDOARJO     SUKODONO     KLOPOSEPULUH       2
3   PROVINSI JAWA TIMUR  SIDOARJO     SUKODONO     KLOPOSEPULUH       3
4   PROVINSI JAWA TIMUR  SIDOARJO     GEDANGAN            SRUNI       4
..                   ...       ...          ...              ...     ...
95  PROVINSI JAWA TIMUR  SIDOARJO        KRIAN        PONOKAWAN       1
96  PROVINSI JAWA TIMUR  SIDOARJO        TARIK        KEDINDING       4
97  PROVINSI JAWA TIMUR  SIDOARJO        TARIK        KEDINDING       5
98  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO  GAGANGKEPUHSARI       2
99  PROVINSI JAWA TIMUR  SIDOARJO  BALONGBENDO  GAGANGKEPUHSARI       5

    suara_1  suara_2  suara_3  golput
0       513       81       29       0
1        72      161      183      10
2       331      620       71       3
3        64      244       39       6
4       343       36      151       0
..      ...      ...      ...     ...
95      295      237      102       1
96       86       68       62       3
97       37       80      128       6
98      209      343      273       3
99       20        4      172       4

[100 rows x 9 columns]
```

Dilakukan pemilihan acak 50 kelurahan dari seluruh data. Dari setiap kelurahan yang terpilih, diambil 2 Tempat Pemungutan Suara (TPS) secara acak. Data dari semua TPS ini kemudian digabungkan menjadi satu *dataframe*.

Sama seperti langkah-langkah sebelumnya, dilakukan estimasi parameter di setiap tingkatan sampel mulai dari 5% hingga 95%.

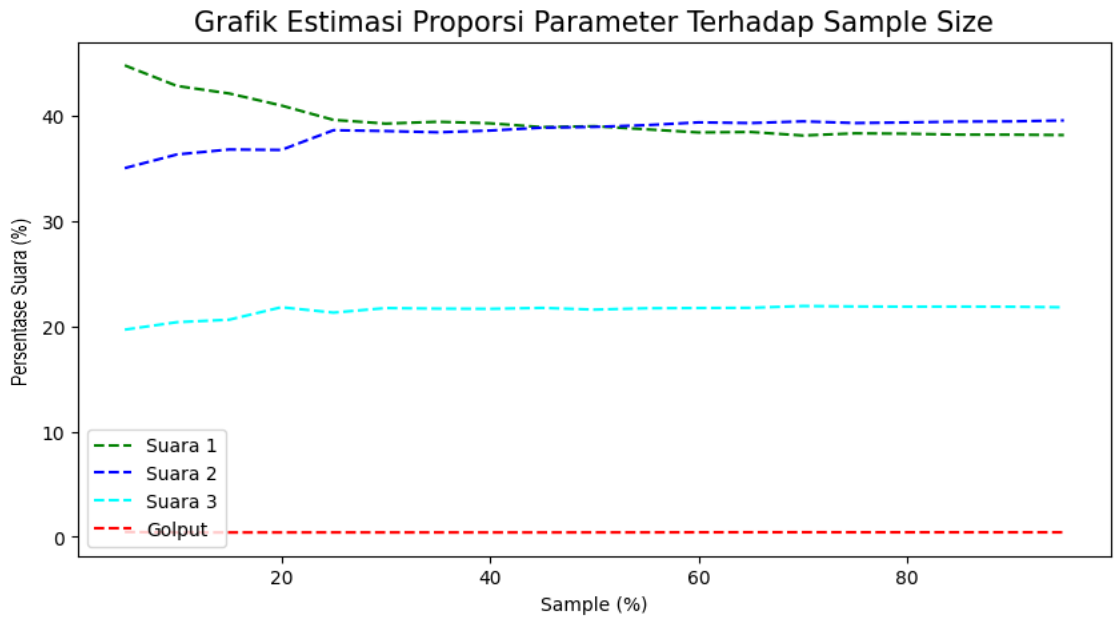
```
# estimasi parameter
estimation_parameter_data = []
estimasi_suara_1 = []
estimasi_suara_2 = []
estimasi_suara_3 = []
estimasi_golput = []

for i in range(1, 20):
  multistage_sampling = completed_data.sample(frac = i / 20, random_st
ate = 42).reset_index(drop = True)
```

```
  # sum all suara_1, suara_2, suara 3, and golput
  suara_1 = multistage_sampling["suara_1"].sum()
  suara_2 = multistage_sampling["suara_2"].sum()
  suara_3 = multistage_sampling["suara_3"].sum()
  golput = multistage_sampling["golput"].sum()

  # append to list
  estimasi_suara_1.append(suara_1)
  estimasi_suara_2.append(suara_2)
  estimasi_golput.append(golput)

  # calculate percentage of suara_1, suara_2, suara 3, and golput
  percentage_suara_1 = (suara_1 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_suara_2 = (suara_2 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_suara_3 = (suara_3 / (suara_1 + suara_2 + suara_3 + golpu
t)) * 100
  percentage_golput = (golput / (suara_1 + suara_2 + suara_3 + golput)
) * 100

  estimation_parameter_data.append({
    "Sample (%)": int(i / 20 * 100),
    "suara_1": percentage_suara_1,
    "suara_2": percentage_suara_2,
    "suara_3": percentage_suara_3,
    "golput": percentage_golput,
  })

# make estimation parameter data to dataframe
estimation_parameter_data = pd.DataFrame(estimation_parameter_data)

print("================== Estimasi Proporsi Parameter ================
==")
estimation_parameter_data
```

```
================== Estimasi Proporsi Parameter ==================

    Sample (%)    suara_1    suara_2    suara_3    golput
0            5  44.803963  35.040043  19.691046  0.464949
1           10  42.847297  36.338491  20.400667  0.413545
2           15  42.135405  36.809342  20.635069  0.420184
3           20  40.990849  36.766572  21.821296  0.421283
4           25  39.621575  38.643144  21.310216  0.425066
..         ...        ...        ...        ...       ...
14          75  38.345705  39.323494  21.894116  0.436685
15          80  38.308014  39.380875  21.875513  0.435598
16          85  38.219195  39.462930  21.882296  0.435580
17          90  38.216661  39.479594  21.868134  0.435611
18          95  38.181046  39.569198  21.816586  0.433170

[19 rows x 5 columns]
```

Hasil estimasi parameter di atas mencakup proporsi suara untuk kandidat-kandidat serta proporsi golput dalam berbagai tingkat persentase sampel. Pada umumnya, terlihat bahwa semakin besar persentase sampel yang diambil, hasil estimasi semakin mendekati nilai sebenarnya, dan fluktuasi semakin berkurang. Fluktuasi yang terjadi dapat dilihat pada grafik di bawah ini.

```
# grafik estimasi parameter terhadap sample size
import matplotlib.pyplot as plt
import seaborn as sns

x = estimation_parameter_data["Sample (%)"]

plt.figure(figsize=(10, 5))
plt.plot(x, estimation_parameter_data["suara_1"], label = "Suara 1", c
olor = "green", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["suara_2"], label = "Suara 2", c
olor = "blue", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["suara_3"], label = "Suara 3", c
olor = "cyan", linestyle = "dashed")
plt.plot(x, estimation_parameter_data["golput"], label = "Golput", col
or = "red", linestyle = "dashed")
plt.title("Grafik Estimasi Proporsi Parameter Terhadap Sample Size", s
ize = 15)
plt.xlabel("Sample (%)")
plt.ylabel("Persentase Suara (%)")
plt.legend()
plt.show()
```



Berdasarkan grafik di atas, dapat dilihat bahwa proporsi suara untuk kandidat pertama (suara_1) cenderung menurun secara perlahan seiring dengan peningkatan persentase sampel. Pada tingkat sampel 5%, suara_1 memiliki proporsi tertinggi sekitar 44.80%, dan pada tingkat sampel 95%, proporsi ini turun menjadi sekitar 38.18%. Proporsi suara untuk kandidat kedua (suara_2) cenderung meningkat secara perlahan seiring dengan peningkatan persentase sampel. Pada tingkat sampel 5%, suara_2 memiliki proporsi sekitar 35.04%, dan pada tingkat sampel 95%, proporsi ini naik menjadi sekitar 39.57%. Proporsi suara untuk kandidat ketiga (suara_3) menunjukkan fluktuasi yang lebih signifikan pada tingkat sampel yang lebih rendah (5-25%). Namun, fluktuasi ini berkurang seiring dengan peningkatan persentase sampel dan stabil di sekitar 21.81-21.88% pada tingkat sampel 85-95%. Proporsi golput cenderung bervariasi pada tingkat sampel yang

lebih rendah (5-25%) dengan nilai tertinggi sekitar 0.46-0.47%. Namun, seiring dengan peningkatan persentase sampel, proporsi golput menjadi lebih stabil di sekitar 0.43-0.44% pada tingkat sampel 75-95%.

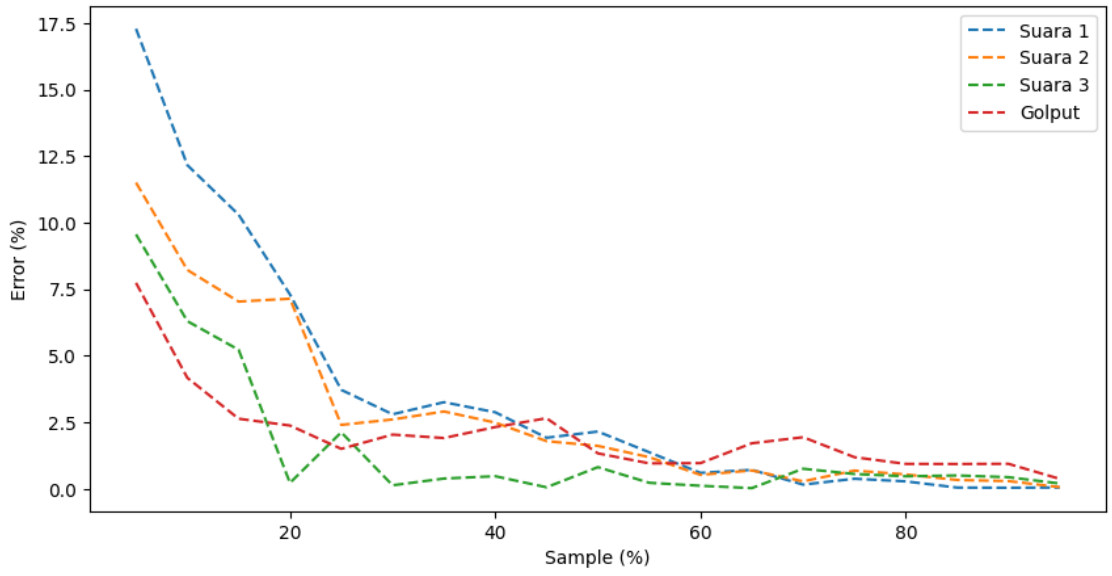
```
# hitung eror estimasi parameter suara_1, suara_2, suara 3 dan golput
percentage_suara_1_kabupaten = (completed_data["suara_1"].sum() / (com
pleted_data["suara_1"].sum() + completed_data["suara_2"].sum()  + comp
leted_data["suara_3"].sum() + completed_data["golput"].sum())) * 100
percentage_suara_2_kabupaten = (completed_data["suara_2"].sum() / (com
pleted_data["suara_1"].sum() + completed_data["suara_2"].sum()  + comp
leted_data["suara_3"].sum() + completed_data["golput"].sum())) * 100
percentage_suara_3_kabupaten = (completed_data["suara_3"].sum() / (com
pleted_data["suara_1"].sum() + completed_data["suara_2"].sum()  + comp
leted_data["suara_3"].sum() + completed_data["golput"].sum())) * 100
percentage_golput_kabupaten = (completed_data["golput"].sum() / (compl
eted_data["suara_1"].sum() + completed_data["suara_2"].sum()  + comple
ted_data["suara_3"].sum() + completed_data["golput"].sum())) * 100

x = estimation_parameter_data["Sample (%)"]
error_suara_1 = abs(100 * (estimation_parameter_data["suara_1"] - perc
entage_suara_1_kabupaten) / percentage_suara_1_kabupaten)
error_suara_2 = abs(100 * (estimation_parameter_data["suara_2"] - perc
entage_suara_2_kabupaten) / percentage_suara_2_kabupaten)
error_suara_3 = abs(100 * (estimation_parameter_data["suara_3"] - perc
entage_suara_3_kabupaten) / percentage_suara_3_kabupaten)
error_golput = abs(100 * (estimation_parameter_data["golput"] - percen
tage_golput_kabupaten) / percentage_golput_kabupaten)

data_error = pd.DataFrame({
  "Sample (%)": x,
  "Error Suara 1": error_suara_1,
  "Error Suara 2": error_suara_2,
  "Error Suara 3": error_suara_3,
  "Error Golput": error_golput
})

data_error.to_csv('output_MCRS.csv', index = False)

print("============== Error Estimasi Proporsi Parameter ==============
")
data_error
```

```
============== Error Estimasi Proporsi Parameter ==============

    Sample (%)  Error Suara 1  Error Suara 2  Error Suara 3  Error Golput
0            5      17.285743      11.504048       9.560753      7.738556
1           10      12.163673       8.224731       6.301526      4.172655
2           15      10.300116       7.035564       5.224941      2.634411
3           20       7.303951       7.143582       0.223296      2.379776
4           25       3.719527       2.404176       2.124050      1.503181
..         ...            ...            ...            ...           ...
14          75       0.379615       0.685908       0.557753      1.189282
15          80       0.280949       0.540987       0.472310      0.937351
16          85       0.048442       0.333751       0.503462      0.933123
17          90       0.041809       0.291666       0.438419      0.940395
18          95       0.051422       0.065366       0.201664      0.374795

[19 rows x 5 columns]
```

Hasil estimasi parameter *error* yang dihasilkan di atas dapat dilihat fluktuasinya pada grafik di bawah ini.

```
# display estimation persentase error of proportion
fig = plt.figure(figsize = (10, 5))

x = estimation_parameter_data["Sample (%)"]

estimation_parameter_data["error_suara_1"] = error_suara_1
estimation_parameter_data["error_suara_2"] = error_suara_2
estimation_parameter_data["error_suara_3"] = error_suara_3
estimation_parameter_data["error_golput"] = error_golput

# graph error
plt.figure(figsize = (10, 5))
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_suara_1"], label = "Suara 1", linestyle = "dashed")
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_suara_2"], label = "Suara 2", linestyle = "dashed")
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_suara_3"], label = "Suara 3", linestyle = "dashed")
plt.plot(estimation_parameter_data["Sample (%)"], estimation_parameter
_data["error_golput"], label = "Golput", linestyle = "dashed")
plt.xlabel("Sample (%)")
plt.ylabel("Error (%)")
plt.legend()
plt.show()
```

```
<Figure size 1000x500 with 0 Axes>
```



Grafik di atas menggambarkan hasil *error* estimasi parameter pada berbagai tingkat persentase sampel. Secara umum, terlihat bahwa *error* estimasi cenderung menurun seiring dengan peningkatan persentase sampel yang diambil. Pada tingkat sampel 5%, terdapat fluktuasi yang signifikan dalam *error* estimasi untuk semua parameter. Namun, seiring dengan peningkatan persentase sampel, fluktuasi ini semakin berkurang, dan *error* estimasi menjadi lebih stabil. Pada tingkat sampel 95%, *error* estimasi secara keseluruhan sangat rendah, dengan fluktuasi yang hampir tidak terlihat. Hal ini menunjukkan bahwa

pengambilan sampel yang lebih besar menghasilkan estimasi parameter yang lebih akurat dan konsisten.

VI. KESIMPULAN

Dalam analisis sampel data untuk memperkirakan hasil Pilkada, berbagai metode *sampling* telah digunakan, termasuk 2 *stage random sampling*, 3 *stage random sampling*, *cluster random sampling*, dan *multistage cluster random sampling*. Hasil estimasi parameter terbaik dengan *error* terendah ditemukan pada metode 3 *stage random sampling*. Pada metode ini, estimasi parameter seperti proporsi suara untuk setiap kandidat dan jumlah golput sangat mendekati hasil sebenarnya. Penggunaan tiga tahap pengambilan sampel yang melibatkan kabupaten, kecamatan, dan kelurahan menghasilkan hasil yang sangat akurat, terutama pada tingkat sampel yang lebih tinggi. *Error* estimasi pada metode ini cenderung rendah dan stabil seiring dengan peningkatan persentase sampel yang diambil.

Sementara itu, metode *cluster random sampling* juga menghasilkan estimasi parameter yang cukup baik, tetapi *error* estimasi pada beberapa tingkat sampel tertentu dapat cukup tinggi. Metode ini memungkinkan pengambilan sampel dalam klaster-klaster tertentu, yang dapat memberikan hasil yang cukup akurat terutama jika klaster-klaster tersebut mewakili variasi yang ada di populasi yang lebih besar.

Selanjutnya, metode *multistage cluster random sampling* menghasilkan estimasi parameter dengan tingkat *error* yang lebih tinggi dibandingkan dengan metode 3 *stage random sampling*. Hal ini terjadi karena pengambilan sampel yang lebih rumit, yang melibatkan beberapa tahap pengambilan sampel dari kelurahan hingga TPS. Meskipun hasilnya cukup akurat, *error* estimasi pada metode ini cenderung lebih tinggi pada tingkat sampel yang lebih rendah.

Secara keseluruhan, untuk mendapatkan hasil estimasi parameter terbaik dan *error* terendah, metode 3 *stage random sampling* merupakan pilihan yang paling baik dalam pemantauan hasil Pilkada. Namun jika dilihat dari fluktuasi *error* yang kian mengerucut seiring naiknya sampel yang digunakan, metode *cluster random sampling* merupakan metode yang memberikan hasil terbaik. Metode ini memberikan tingkat akurasi yang tinggi, terutama ketika lebih banyak sampel diambil, sehingga hasilnya dapat diandalkan untuk menggambarkan hasil Pilkada pada tingkat kabupaten dengan tingkat kesalahan yang minimal.

VII. DAFTAR PUSTAKA

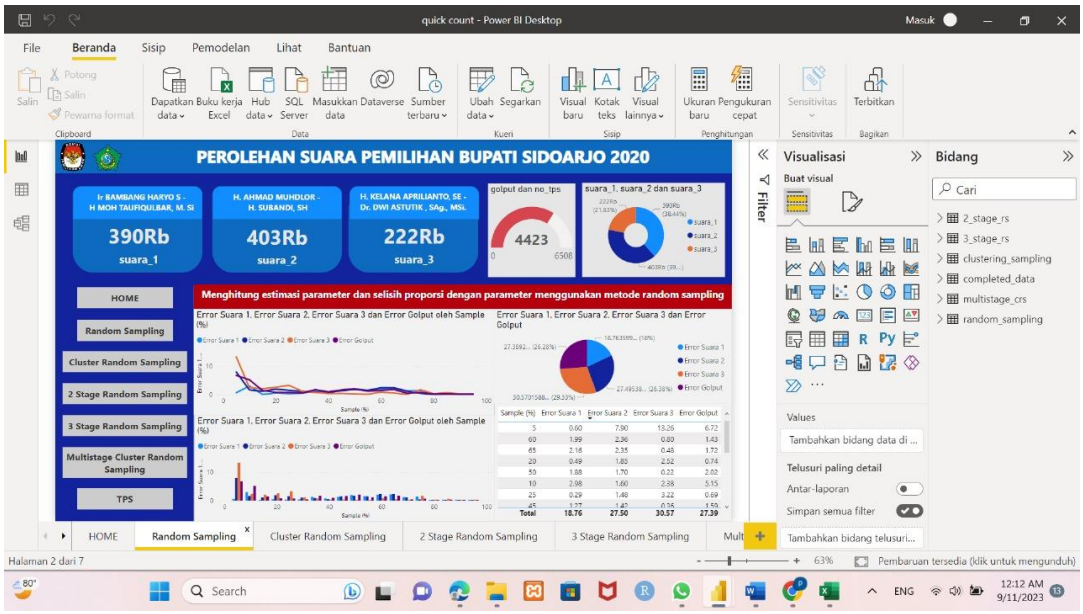
[1] Desvira, D. Penerapan Algoritma Greedy dalam Menentukan Sampel TPS pada Quick Count. *Makalah IF221 Strategi Algoritma, Program Studi Teknik Informatika.* Institut Teknologi Bandung, 2014.

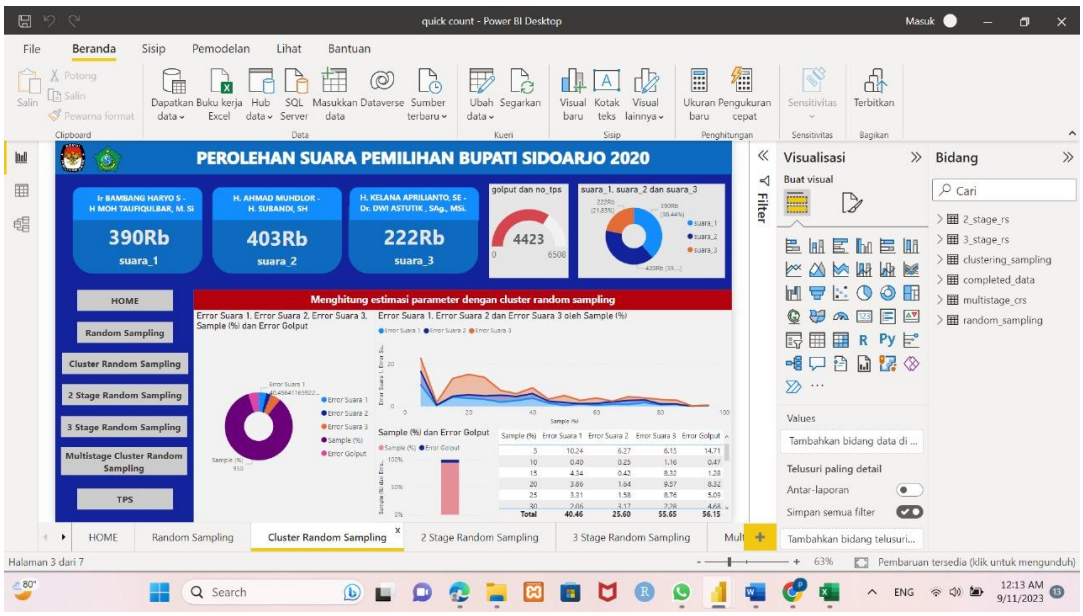
[2] Nurdin, Hamdhana, D. & Iqbal, M. Aplikasi Quick Count Pilkada dengan Menggunakan Metode Random Sampling Berbasis Android. *Program Studi Teknik Informatika*. Universitas Malikussaleh.

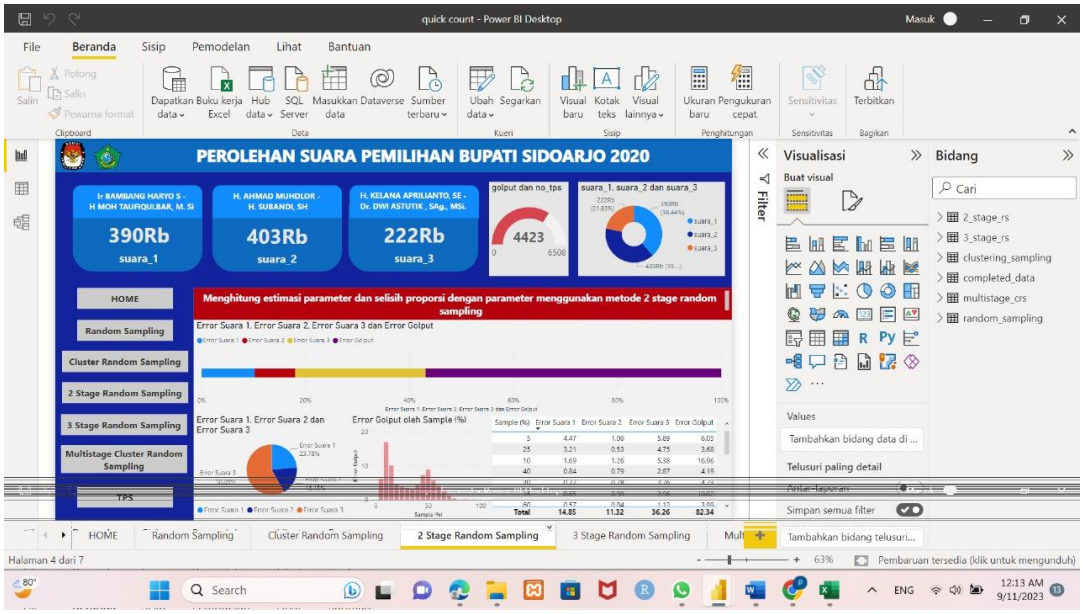
[3] Saputra, A. Y. & Apriadi, D. Rancang Bangun Aplikasi Quick Count Pilkada Berbasis SMS Gateway dengan Metode Simple Random Sampling (Studi Kasus Kota Lubuklingau). *Journal Information System Development (ISD)*. Vol. 3, No. 1, 2018.

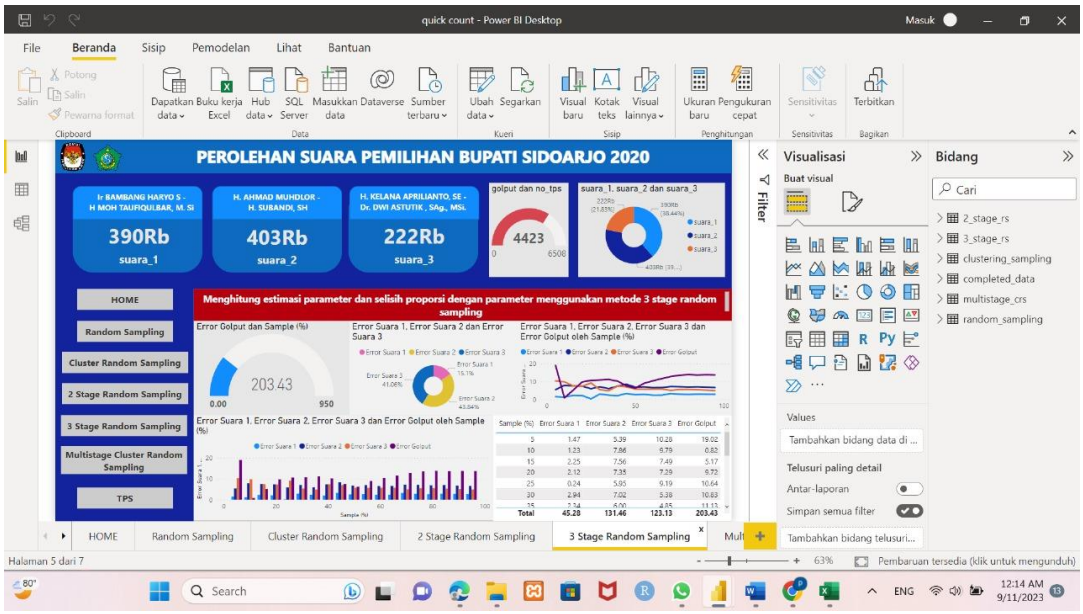
VIII. LAMPIRAN

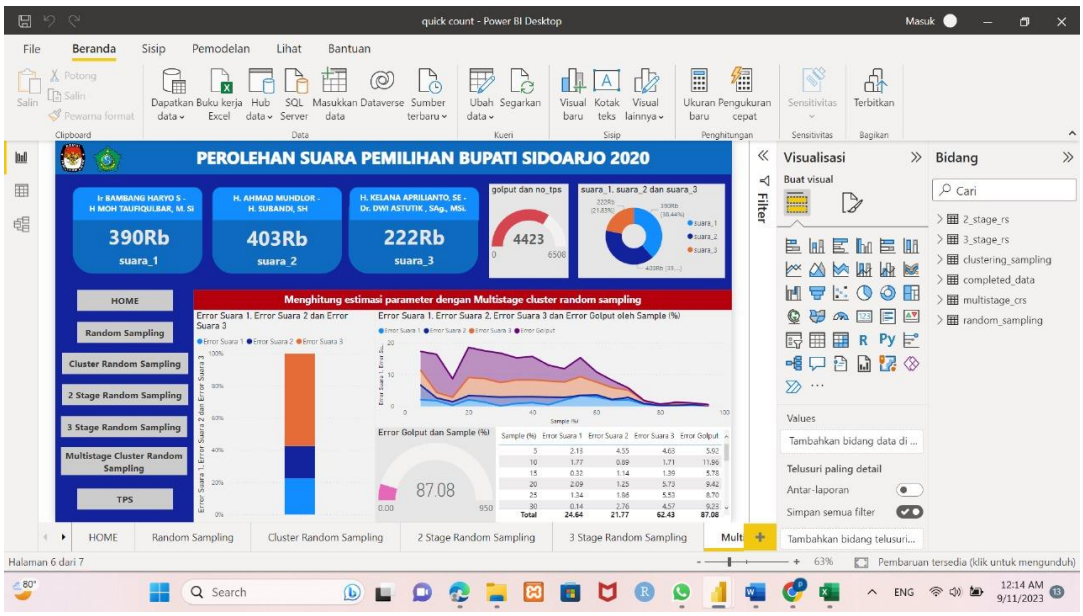
Lampiran tampilan *dashboard*

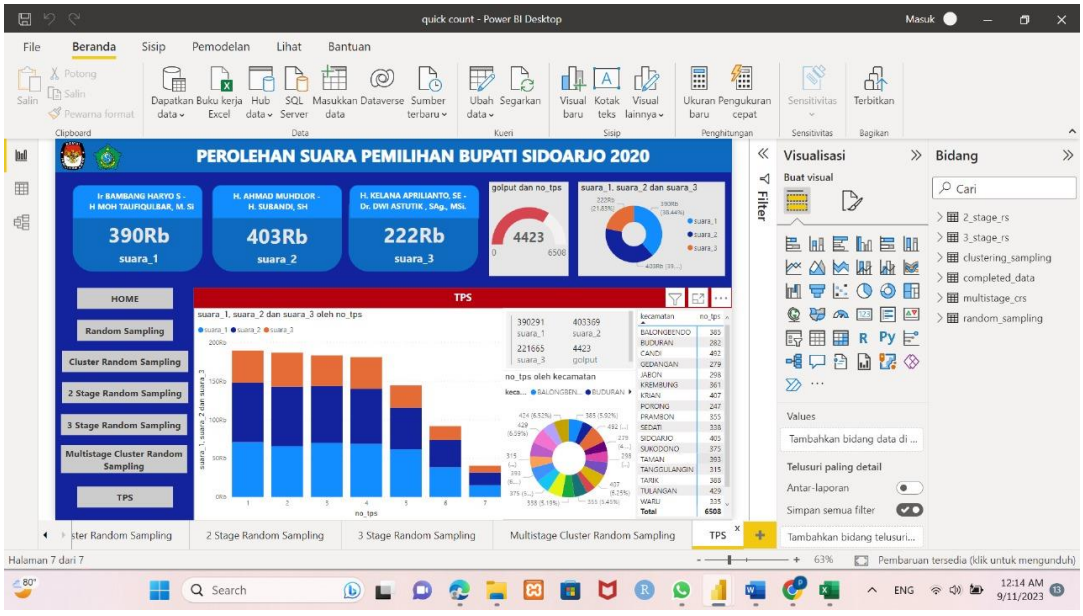



quick count - Power BI Desktop
PEROLEHAN SUARA PEMILIHAN BUPATI SIDOARJO 2020
Ir BAMBANG HARYO S - H MOH TAUFIQULBAR, M. SI
390Rb
suara_1
H. AHMAD MUHDLOR - H. SUBANDI, SH
403Rb
suara_2
H. KELANA APRILIANTO, SE - Dr. DWI ASTUTIK , SAg., MSi.
222Rb
suara_3
golput dan no_tps
4423
suara_1, suara_2 dan suara_3
HOME
Random Sampling
Cluster Random Sampling
2 Stage Random Sampling
3 Stage Random Sampling
Multistage Cluster Random Sampling
TPS
Menghitung estimasi parameter dengan cluster random sampling
Error Suara 1, Error Suara 2, Error Suara 3, Sample (%) dan Error Golput
Error Suara 1, Error Suara 2 dan Error Suara 3 oleh Sample (%)
Sample (%) dan Error Golput
Visualisasi
Buat visual
Bidang
Cari
2_stage_rs
3_stage_rs
clustering_sampling
completed_data
multistage_crs
random_sampling
Values
Tambahkan bidang data di ...
Telusuri paling detail
Antar-laporan
Simpan semua filter
Tambahkan bidang telusuri...
Filter
HOME
Random Sampling
Cluster Random Sampling
2 Stage Random Sampling
3 Stage Random Sampling
Halaman 3 dari 7
63%
Pembaruan tersedia (klik untuk mengunduh)
Search
ENG
12:13 AM
9/11/2023


quick count - Power BI Desktop
PEROLEHAN SUARA PEMILIHAN BUPATI SIDOARJO 2020
Ir BAMBANG HARYO S - H MOH TAUFIQULBAR, M. SI
390Rb
suara_1
H. AHMAD MUHDLOR - H. SUBANDI, SH
403Rb
suara_2
H. KELANA APRILIANTO, SE - Dr. DWI ASTUTIK , SAg., MSi.
222Rb
suara_3
golput dan no_tps
4423
suara_1, suara_2 dan suara_3
HOME
Random Sampling
Cluster Random Sampling
2 Stage Random Sampling
3 Stage Random Sampling
Multistage Cluster Random Sampling
TPS
Menghitung estimasi parameter dan selisih proporsi dengan parameter menggunakan metode 2 stage random sampling
Error Suara 1, Error Suara 2, Error Suara 3 dan Error Golput
Error Suara 1, Error Suara 2 dan Error Suara 3
Error Golput oleh Sample (%)
Visualisasi
Buat visual
Bidang
Cari
2_stage_rs
3_stage_rs
clustering_sampling
completed_data
multistage_crs
random_sampling
Values
Tambahkan bidang data di ...
Telusuri paling detail
Antar-laporan
Simpan semua filter
Tambahkan bidang telusuri...
Filter
HOME
Random Sampling
Cluster Random Sampling
2 Stage Random Sampling
3 Stage Random Sampling
Halaman 4 dari 7
63%
Pembaruan tersedia (klik untuk mengunduh)
Search
ENG
12:13 AM
9/11/2023


quick count - Power BI Desktop
PEROLEHAN SUARA PEMILIHAN BUPATI SIDOARJO 2020
Ir BAMBANG HARYO S - H MOH TAUFIQULBAR, M. SI
390Rb
suara_1
H. AHMAD MUHDLOR - H. SUBANDI, SH
403Rb
suara_2
H. KELANA APRILIANTO, SE - Dr. DWI ASTUTIK , SAg., MSi.
222Rb
suara_3
golput dan no_tps
4423
suara_1, suara_2 dan suara_3
HOME
Random Sampling
Cluster Random Sampling
2 Stage Random Sampling
3 Stage Random Sampling
Multistage Cluster Random Sampling
TPS
Menghitung estimasi parameter dan selisih proporsi dengan parameter menggunakan metode 3 stage random sampling
Error Golput dan Sample (%)
Error Suara 1, Error Suara 2 dan Error Suara 3
Error Suara 1, Error Suara 2, Error Suara 3 dan Error Golput oleh Sample (%)
Error Suara 1, Error Suara 2, Error Suara 3 dan Error Golput oleh Sample (%)
Visualisasi
Buat visual
Bidang
Cari
2_stage_rs
3_stage_rs
clustering_sampling
completed_data
multistage_crs
random_sampling
Values
Tambahkan bidang data di ...
Telusuri paling detail
Antar-laporan
Simpan semua filter
Tambahkan bidang telusuri...
Filter
HOME
Random Sampling
Cluster Random Sampling
2 Stage Random Sampling
3 Stage Random Sampling
Halaman 5 dari 7
63%
Pembaruan tersedia (klik untuk mengunduh)
Search
ENG
12:14 AM
9/11/2023


quick count - Power BI Desktop
PEROLEHAN SUARA PEMILIHAN BUPATI SIDOARJO 2020
Ir BAMBANG HARYO S - H MOH TAUFIQULBAR, M. SI
390Rb
suara_1
H. AHMAD MUHDLOR - H. SUBANDI, SH
403Rb
suara_2
H. KELANA APRILIANTO, SE - Dr. DWI ASTUTIK , SAg., MSi.
222Rb
suara_3
golput dan no_tps
4423
suara_1, suara_2 dan suara_3
HOME
Random Sampling
Cluster Random Sampling
2 Stage Random Sampling
3 Stage Random Sampling
Multistage Cluster Random Sampling
TPS
Menghitung estimasi parameter dengan Multistage cluster random sampling
Error Suara 1, Error Suara 2 dan Error Suara 3
Error Suara 1, Error Suara 2, Error Suara 3 dan Error Golput oleh Sample (%)
Error Golput dan Sample (%)
87.08
Visualisasi
Buat visual
Bidang
Cari
2_stage_rs
3_stage_rs
clustering_sampling
completed_data
multistage_crs
random_sampling
Values
Tambahkan bidang data di ...
Telusuri paling detail
Antar-laporan
Simpan semua filter
Tambahkan bidang telusuri...
Halaman 6 dari 7
63%
Pembaruan tersedia (klik untuk mengunduh)
Search
ENG
12:14 AM
9/11/2023


quick count - Power BI Desktop
PEROLEHAN SUARA PEMILIHAN BUPATI SIDOARJO 2020
Ir BAMBANG HARYO S - H MOH TAUFIQULBAR, M. SI
390Rb
suara_1
H. AHMAD MUHDLOR - H. SUBANDI, SH
403Rb
suara_2
H. KELANA APRILIANTO, SE - Dr. DWI ASTUTIK , SAg., MSi.
222Rb
suara_3
golput dan no_tps
4423
suara_1, suara_2 dan suara_3
HOME
Random Sampling
Cluster Random Sampling
2 Stage Random Sampling
3 Stage Random Sampling
Multistage Cluster Random Sampling
TPS
TPS
suara_1, suara_2 dan suara_3 oleh no_tps
390291
suara_1
403369
suara_2
221665
suara_3
4423
golput
no_tps oleh kecamatan
kecamatan
no_tps
BALONGBENDO
BUDURAN
CANDI
GEDANGAN
JABON
KREMBUNG
KRIAN
PORONG
PRAMBON
SEDATI
SIDOARJO
SUKODONO
TAMAN
TANGGULANGIN
TARIK
TULANGAN
WARU
Total
6508
Visualisasi
Buat visual
Bidang
Cari
2_stage_rs
3_stage_rs
clustering_sampling
completed_data
multistage_crs
random_sampling
Values
Tambahkan bidang data di ...
Telusuri paling detail
Antar-laporan
Simpan semua filter
Tambahkan bidang telusuri...
Halaman 7 dari 7
63%
Pembaruan tersedia (klik untuk mengunduh)
Search
ENG
12:14 AM
9/11/2023