



PRAKTIKUM ANALISA STATISTIKA TERAPAN

**MEMBANGUN APLIKASI *QUICK COUNT* PEMILIHAN PRESIDEN
TAHUN 2019 PADA PROVINSI KALIMANTAN TENGAH**

SAINS DATA TERAPAN

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



ANGGOTA KELOMPOK 10:

3322600009

Dicky Syarif Renaldi

3322600023

Johan Winarwan Nawawi

3322600025

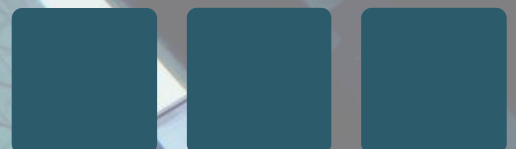
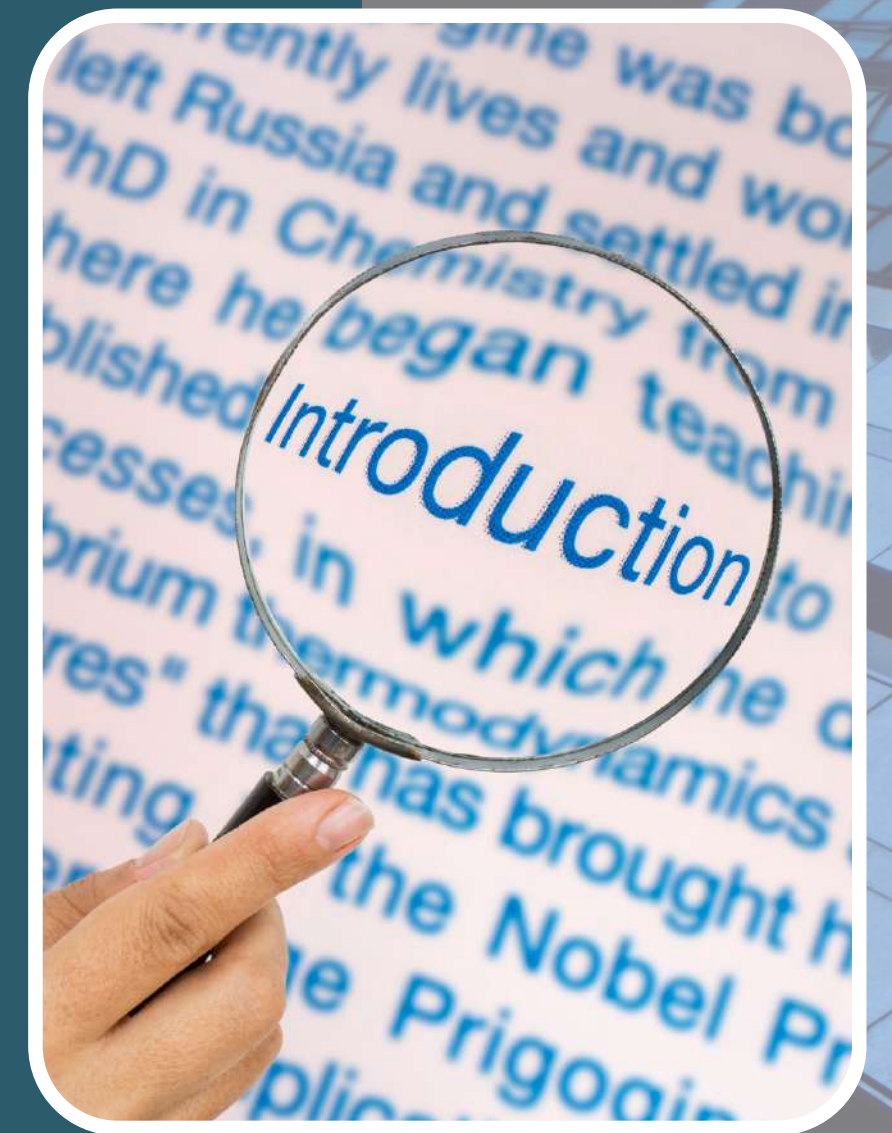
Eky Fernanda Setyawan P



LATAR BELAKANG

Indonesia adalah negara demokratis yang menekankan kedaulatan rakyat. Pemilihan umum (Pemilu) adalah ukuran utama dalam menilai kesehatan demokrasi di Indonesia. Pemilihan Presiden (Pilpres) adalah elemen penting dalam sistem demokrasi Indonesia. Namun, pelaksanaan Pemilu di Indonesia, dengan kompleksitas geografisnya, seringkali memerlukan waktu yang lama untuk mengumumkan hasil perolehan suara, yang dapat mengakibatkan kesulitan dalam memantau dinamika dukungan calon dan potensi manipulasi hasil.

Sejak tahun 2004, metode *quick count* atau hitung cepat hadir sebagai upaya untuk mengatasi keterlambatan ini. *Quick count* adalah metode verifikasi hasil pemilihan umum yang menghitung persentase hasil pemilu di Tempat Pemungutan Suara (TPS) yang dijadikan sampel. Proses ini dilakukan oleh lembaga survei untuk mendapatkan hasil sementara sebelum hasil resmi dikeluarkan oleh Komisi Pemilihan Umum (KPU). *Quick count* dilakukan dengan sampel yang representatif dan memiliki tingkat akurasi tinggi, dan ini membantu menjaga integritas proses demokratis serta memastikan hasil yang adil dan jujur. Metode ini juga digunakan untuk membandingkan hasil dengan hasil resmi KPU.



TUJUAN

Tujuan praktikum ini adalah untuk mengembangkan kemampuan mahasiswa dalam berbagai aspek pengambilan sampel dan analisis data terkait pemilihan presiden (Pilpres). Adapun tujuan khususnya adalah:

- Implementasi algoritma *generate data* dengan distribusi *Dirichlet*;
- Implementasi *simple random sampling* dalam pengambilan sampel;
- Implementasi metode *clustering random sampling* dan *multistage random sampling*;
- Membangun sebuah algoritma sistem *quick count* Pilpres;
- Melakukan perbandingan dan evaluasi estimator (parameter proporsi) dari beberapa metode *sampling*; serta
- Membangun aplikasi simulasi *quick count* Pilpres dengan bahasa pemrograman R/*Jupyter Notebook* atau bahasa pemrograman yang lain.



DATASET



Sumber Data

Hasil Pilpres di Provinsi Kalimantan Tengah Tahun 2019

Pemilu Presiden RI di Provinsi Kalimantan Tengah Tahun 2019

<https://pemilu2019.kpu.go.id/#/ppwp/hitung-suara/>

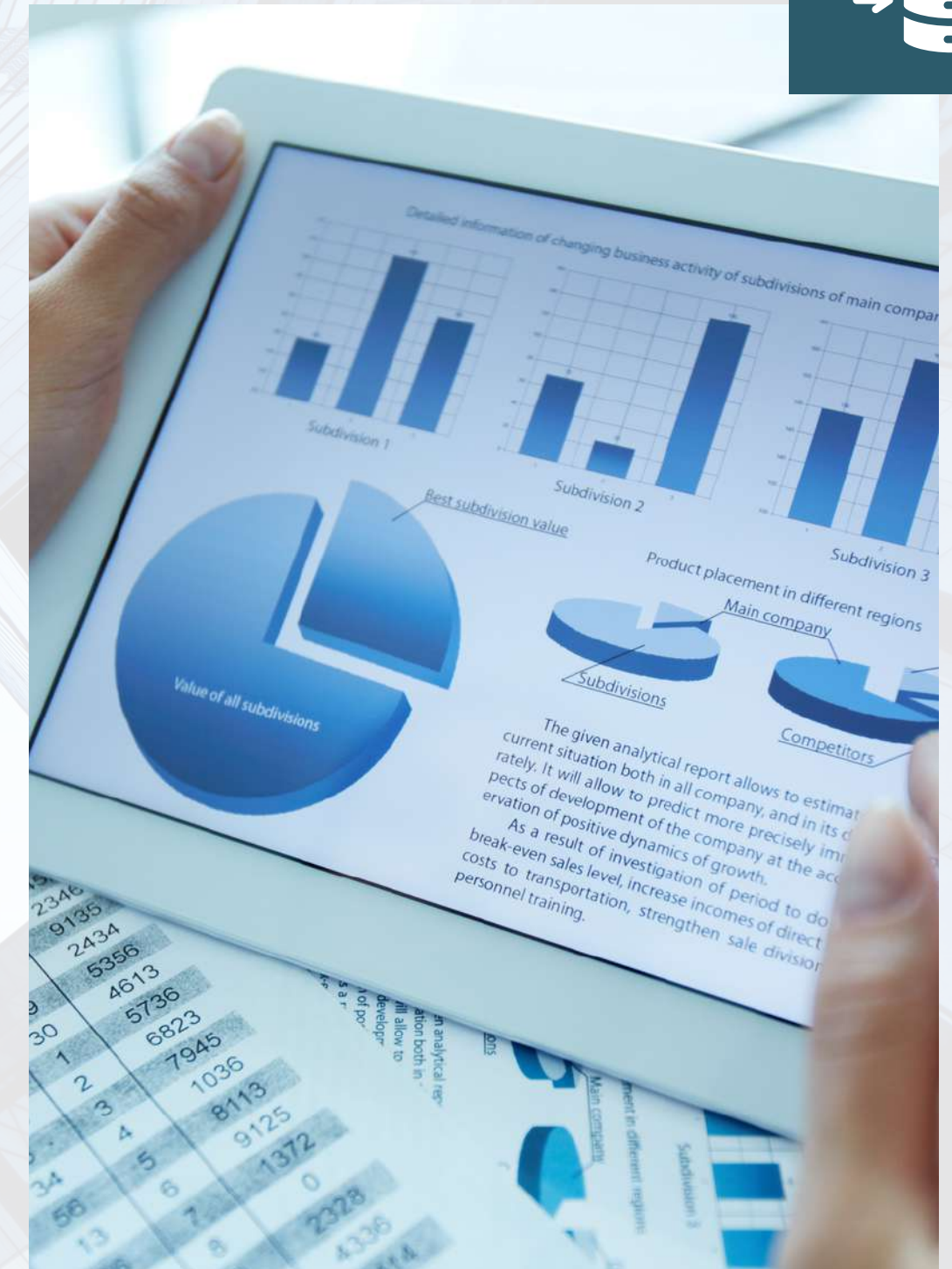
Pilpres Kalteng

Dimensi data Pilpres Kalteng 1572 x 6

<https://intip.in/pilpreskaltengtahun2019>



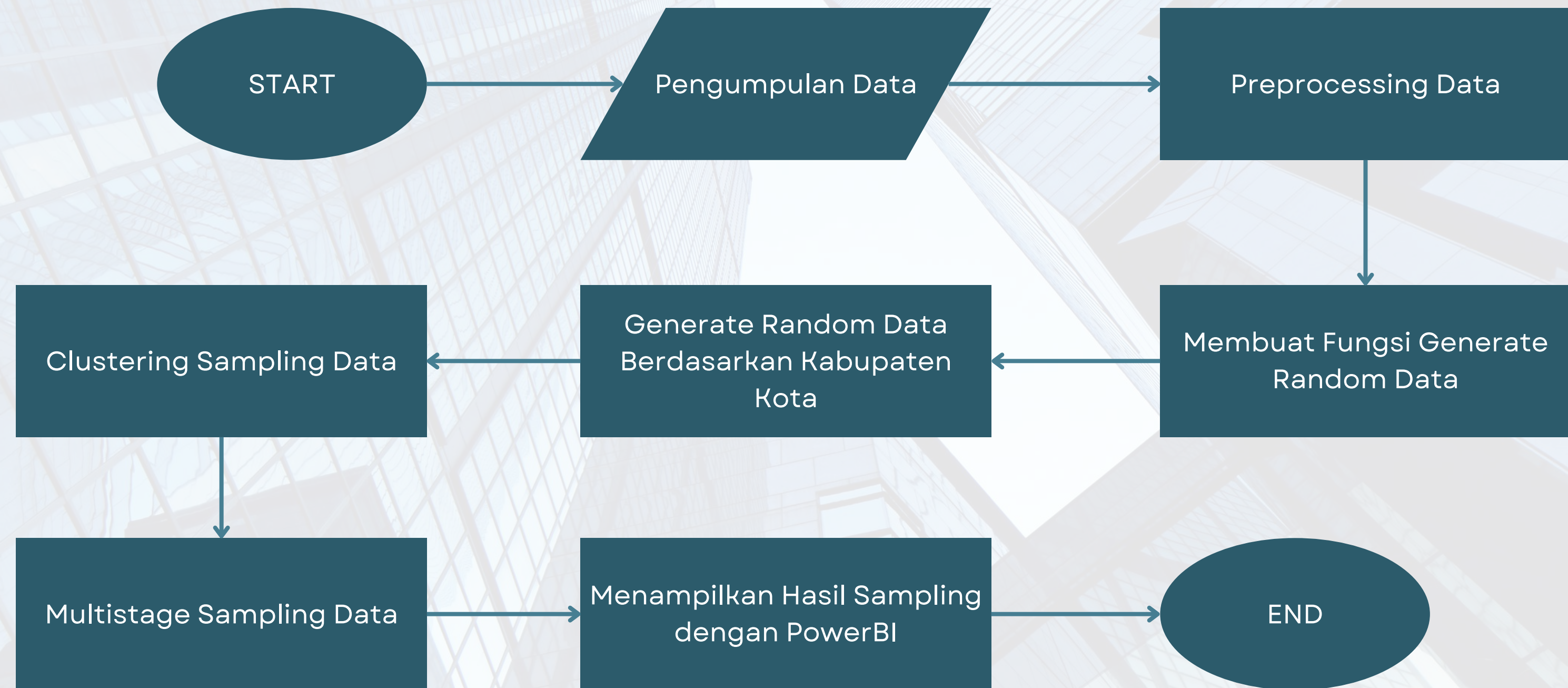
Dataset Digunakan



Variabel	Deskripsi
Provinsi	Berisikan provinsi Kalimantan Tengah.
Kabupaten_Kota	Daftar kabupaten/kota di Kalimantan Tengah.
Kecamatan	Daftar nama kecamatan sesuai kabupaten/kota
Kelurahan	Daftar nama kelurahan sesuai kecamatan
Jokowi	Total hasil suara pemilih Jokowi tiap Kelurahan.
Prabowo	Total hasil suara pemilih Prabowo tiap Kelurahan.



METODOLOGI



Flowchart Metodologi





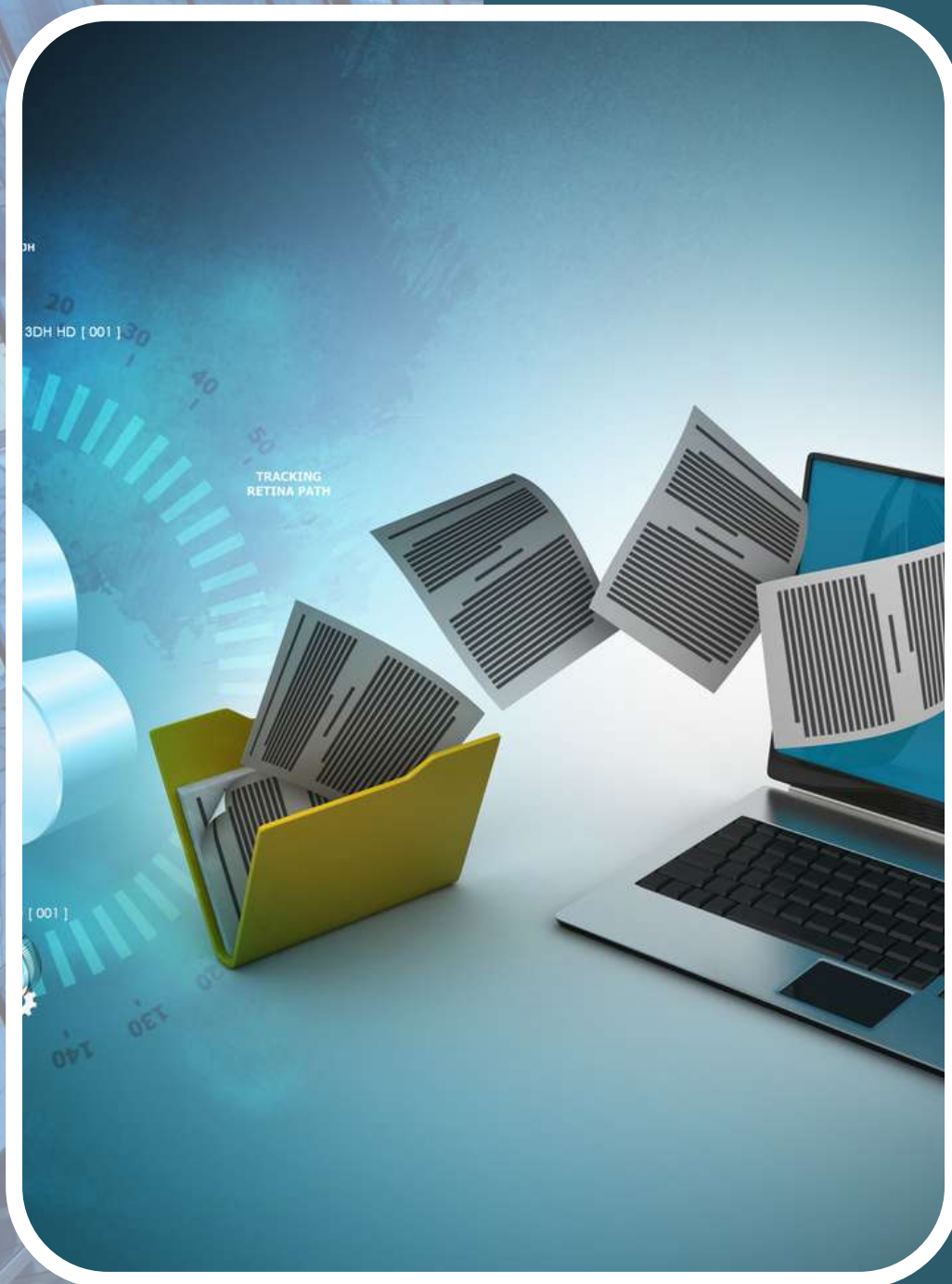
HASIL & PEMBAHASAN

- Pengumpulan Data Pilpres Provinsi Kalimantan Tengah Tahun 2019

- Estimasi Parameter dan Selisih Proporsi dengan Parameter Menggunakan Metode *Random Sampling*

- Estimasi Parameter dengan *Cluster Random Sampling*

- Estimasi Parameter dengan *Multistage Random Sampling*



Pengumpulan Data Pilpres Provinsi Kalimantan Tengah Tahun 2019



Pengumpulan Data Pilpres Provinsi Kalimantan Tengah Tahun 2019

	Provinsi	Kabupaten_Kota	Kecamatan	Kelurahan	Jokowi	Prabowo
0	Kalimantan Tengah	Barito Selatan	Dusun Hilir	Batampang	469	424
1	Kalimantan Tengah	Barito Selatan	Dusun Hilir	Batilap	328	113
2	Kalimantan Tengah	Barito Selatan	Dusun Hilir	Damparan	685	770
3	Kalimantan Tengah	Barito Selatan	Dusun Hilir	Kalanis	387	1197
4	Kalimantan Tengah	Barito Selatan	Dusun Hilir	Lehai	168	661
...
1567	Kalimantan Tengah	Sukamara	Sukamara	Padang	2005	1512
1568	Kalimantan Tengah	Sukamara	Sukamara	Pangkalan Muntai	731	488
1569	Kalimantan Tengah	Sukamara	Sukamara	Petarikan	379	145
1570	Kalimantan Tengah	Sukamara	Sukamara	Pudu	426	135
1571	Kalimantan Tengah	Sukamara	Sukamara	Sukaraja	243	185

1572 rows x 6 columns

Data yang diperlukan mencakup informasi dari berbagai kabupaten/kota di provinsi Kalimantan Tengah. Dalam proses ini data tingkat yang lebih detail seperti kecamatan dan kelurahan juga diambil karena merupakan bagian integral. Panjang data yang diperoleh yakni 1572 baris data dengan 6 kolom.

Generate Tempat Pemungutan Suara (TPS) dengan Menggunakan Random Dirichlet

```
def generate_tps(data, suara1, suara2, golput):-
    kelurahan = data["Kelurahan"].unique()
    kelurahan_tps = []
    for kel in kelurahan:
        kelurahan_tps.append({
            "kelurahan": kel,
            "jumlah_tps": random.randint(3, 7)
        })

    tps = []
    for kel in kelurahan_tps:
        for i in range(kel["jumlah_tps"]):
            tps.append({
                "kelurahan": kel["kelurahan"],
                "nomor_tps": i+1,
                "suara1": 0,
                "suara2": 0,
                "golput": 0
            })

    random_suara1 = (np.random.dirichlet(np.ones(len(tps)),size=1)[0] * suara1).tolist()
    random_suara2 = (np.random.dirichlet(np.ones(len(tps)),size=1)[0] * suara2).tolist()
    random_golput = (np.random.dirichlet(np.ones(len(tps)),size=1)[0] * golput).tolist()

    random_suara1 = [round(i) for i in random_suara1]
    random_suara2 = [round(i) for i in random_suara2]
    random_golput = [round(i) for i in random_golput]
```

```
for i in range(len(tps)):
    tps[i]["suara1"] = random_suara1[i]
    tps[i]["suara2"] = random_suara2[i]
    tps[i]["golput"] = random_golput[i]

tps.append({
    "kelurahan": "TOTAL",
    "nomor_tps": sum([kel["jumlah_tps"] for kel in kelurahan_tps]),
    "suara1": sum([t["suara1"] for t in tps]),
    "suara2": sum([t["suara2"] for t in tps]),
    "golput": sum([t["golput"] for t in tps])
})

return pd.DataFrame(tps)
```

Fungsi **generate_tps** merupakan alat untuk menghasilkan data simulasi hasil suara pemilihan presiden di berbagai Tempat Pemungutan Suara (TPS) dan kelurahan. Pendekatan ini memungkinkan simulasi hasil suara dengan memperhitungkan jumlah TPS per kelurahan dan proporsi suara untuk masing-masing kandidat serta golongan putih. Penggunaan distribusi *Dirichlet* membantu membangkitkan proporsi suara acak yang realistis untuk setiap TPS, dengan hasil suara yang dibulatkan ke bilangan bulat terdekat. Tujuan utama adalah memfasilitasi eksperimen, pelatihan, analisis sensitivitas, dan pengujian algoritma terkait perhitungan cepat (*quick count*) dalam pemilihan umum, membantu pemahaman tentang variasi hasil suara yang mungkin terjadi di setiap unit pemungutan suara.



Simpan Hasil *Generate* TPS Menjadi *DataFrame* Baru

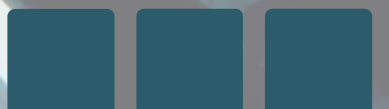
```
# collect all data suara and remove total row from every data
tps = pd.concat([tps_barito_selatan, tps_barito_timur, tps_barito_utara, tps_gunung_mas, tps_kapuas, tps_katingan,
                tps_kota_palangkaraya, tps_kotawaringin_barat, tps_kotawaringin_timur, tps_lamandau, tps_murung_raya,
                tps_pulang_pisau, tps_seruyan, tps_sukamara])

tps = tps[tps["kelurahan"] != "TOTAL"]
tps = tps.rename(columns={'kelurahan': 'Kelurahan'})

total_row = pd.DataFrame([{"Provinsi": "TOTAL",
                           "Kabupaten_Kota": len(data["Kabupaten_Kota"].unique()),
                           "Kecamatan": len(data["Kecamatan"].unique()),
                           "Kelurahan": len(tps["Kelurahan"].unique()),
                           "nomor_tps": len(tps["Kelurahan"].unique()) * len(tps["nomor_tps"].unique()),
                           "suara1": tps["suara1"].sum(),
                           "suara2": tps["suara2"].sum(),
                           "golput": tps["golput"].sum()}])

# concat data suara with data
completed_data = pd.merge(data, tps, on="Kelurahan", how="outer")
completed_data = pd.concat([completed_data, total_row], ignore_index=True)
completed_data.to_excel("D:/PENS/Semester 3/Analisa Statistika Terapan/completed_data.xlsx", index=False)
```

Setelah semua kabupaten/kota dilakukan generate TPS, maka hasil dari *generate* tersebut akan disimpan dalam *dataframe* yang akan digunakan untuk analisis lebih lanjut.



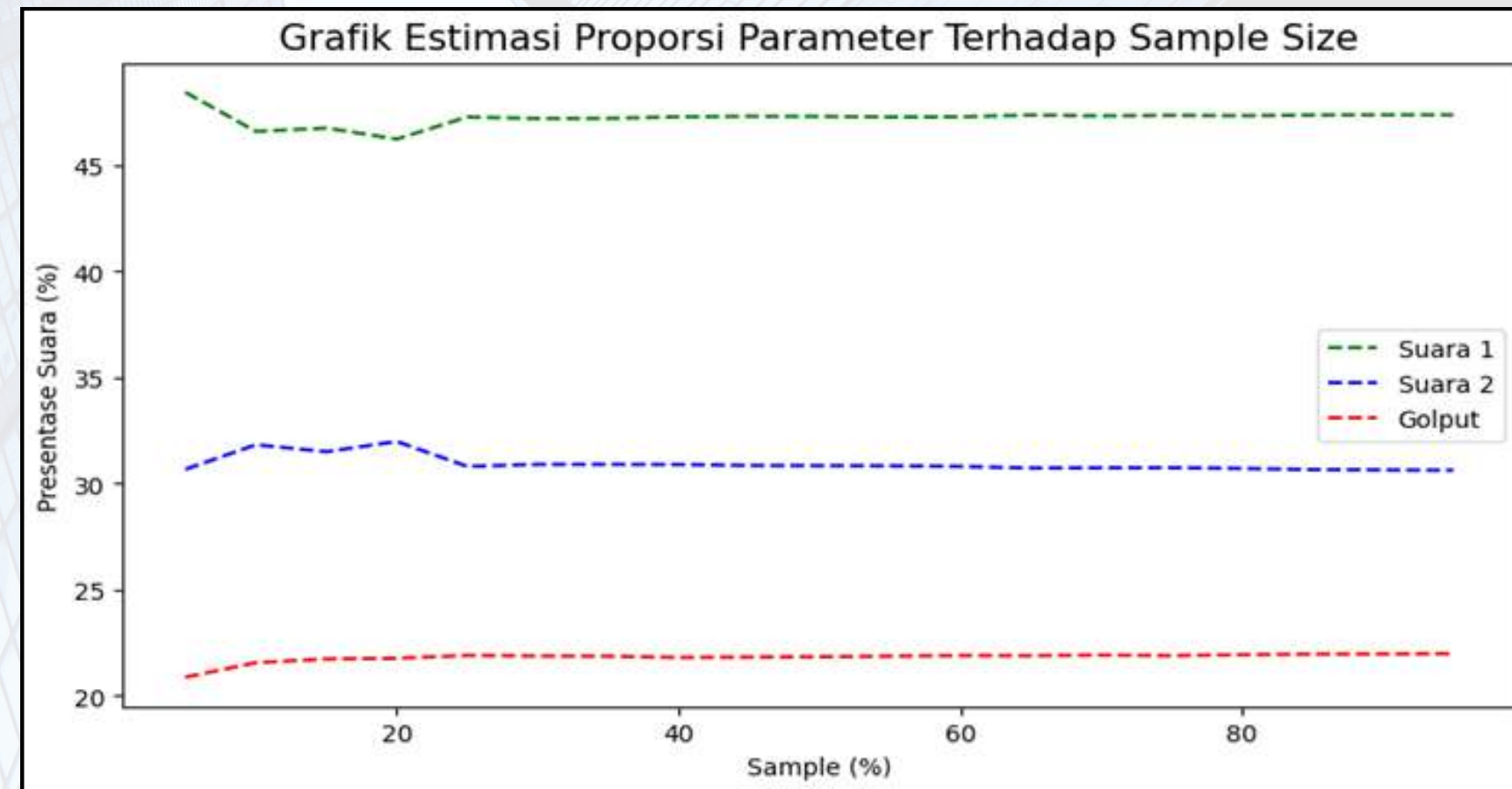


Estimasi Parameter dan Selisih Proporsi dengan Parameter Menggunakan Metode *Random Sampling*



Estimasi Parameter dan Selisih Proporsi dengan Parameter Menggunakan Metode *Random Sampling*

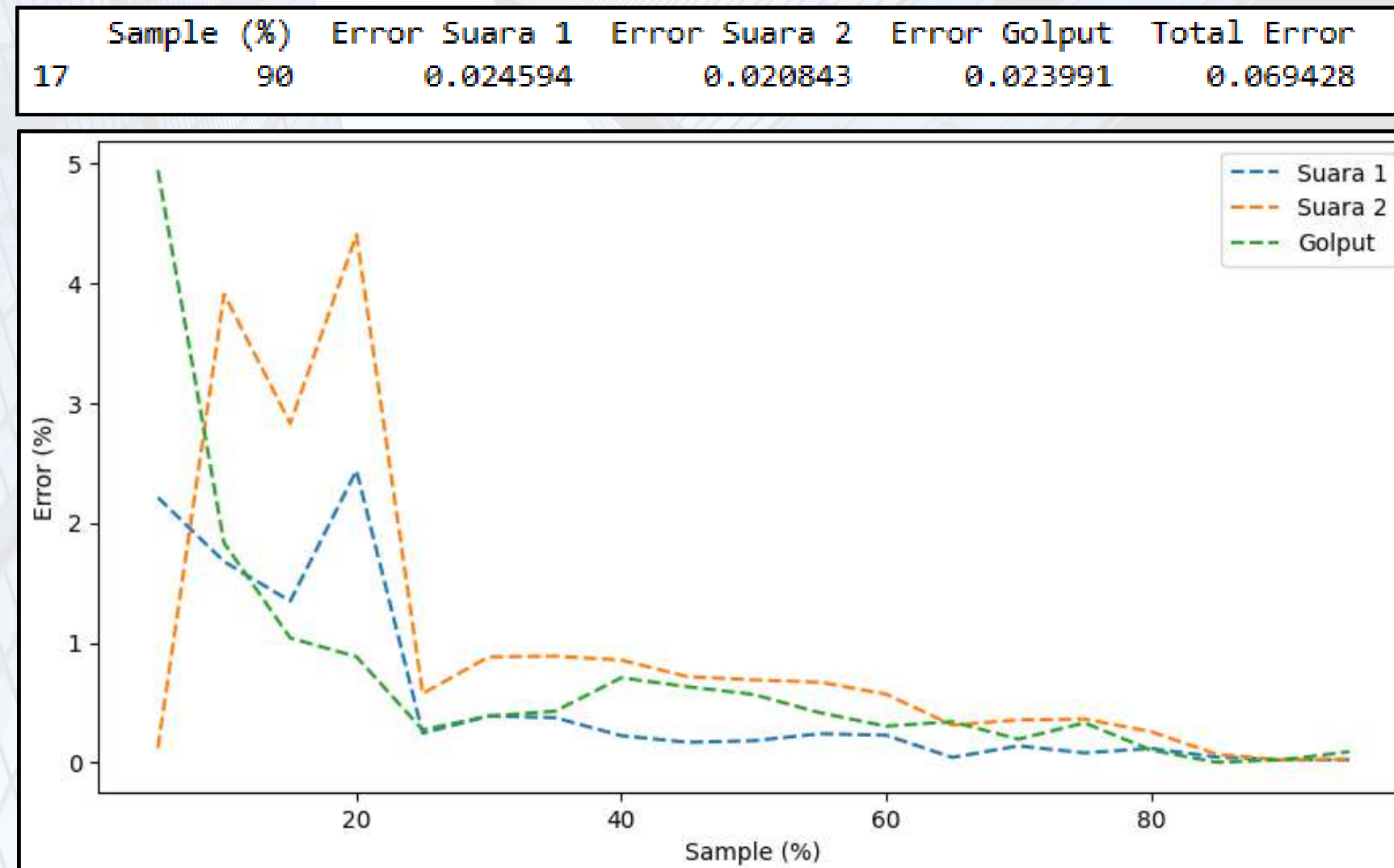
=====Estimasi Proporsi Parameter=====				
	Sample (%)	suara1	suara2	golput
0	5	48.442278	30.676201	20.881521
1	10	46.599139	31.836424	21.564437
2	15	46.755590	31.505713	21.738696
3	20	46.238846	31.988325	21.772830
4	25	47.277412	30.815617	21.906971
5	30	47.208946	30.909452	21.881602
6	35	47.216088	30.911207	21.872705
7	40	47.287278	30.901176	21.811545
8	45	47.313195	30.858954	21.827852
9	50	47.307255	30.850705	21.842040
10	55	47.279830	30.844653	21.875517
11	60	47.285486	30.814345	21.900169
12	65	47.373152	30.734936	21.891912
13	70	47.327700	30.748260	21.924040
14	75	47.355371	30.750598	21.894031
15	80	47.337681	30.718447	21.943872
16	85	47.371855	30.660487	21.967658
17	90	47.382263	30.645451	21.972286
18	95	47.382279	30.630675	21.987045



Berdasarkan 19 iterasi dengan fraksi sampel dari 5% hingga 95% dari data asli, estimasi persentase suara1 mendekati 50% pada awal, namun mulai berfluktuasi seiring dengan variasi sampel yang lebih besar. Suara2 awalnya mendekati 35% dan juga mengalami fluktuasi. Persentase golput stabil di sekitar 20% pada awal, naik sedikit ketika sampel lebih besar. Semakin besar sampel, estimasi lebih stabil dan mendekati nilai sebenarnya, mengikuti hukum statistik yang menyatakan bahwa sampel yang lebih besar memberikan estimasi yang lebih akurat.

Melakukan Evaluasi Terhadap Tingkat Kesalahan (*Error*) dari Estimasi Metode *Random Sampling*

=====Error Estimasi Proporsi Parameter=====				
	Sample (%)	Error Suara 1	Error Suara 2	Error Golput
0	5	2.212011	0.121205	4.941475
1	10	1.676966	3.907950	1.832653
2	15	1.346858	2.828573	1.039374
3	20	2.437177	4.403724	0.883990
4	25	0.245828	0.576232	0.273339
5	30	0.390290	0.882493	0.388829
6	35	0.375219	0.888219	0.429329
7	40	0.225010	0.855481	0.707746
8	45	0.170327	0.717674	0.633513
9	50	0.182860	0.690753	0.568925
10	55	0.240725	0.671001	0.416530
11	60	0.228792	0.572082	0.304307
12	65	0.043818	0.312906	0.341896
13	70	0.139720	0.356392	0.195640
14	75	0.081337	0.364023	0.332247
15	80	0.118662	0.259088	0.105356
16	85	0.046554	0.069916	0.002924
17	90	0.024594	0.020843	0.023991
18	95	0.024560	0.027381	0.091179



Semakin besar sampel, semakin kecil *error* estimasi. Total *error* dihitung untuk setiap baris data, dan baris dengan total *error* terkecil diidentifikasi. Dalam grafik, perubahan tingkat *error* estimasi suara1, suara2, dan golput dengan ukuran sampel terlihat. Awalnya, *error* tinggi (sekitar 5%) pada sampel kecil, mengindikasikan ketidakakuratan estimasi. Namun, dengan ukuran sampel yang bertambah, tingkat *error* secara bertahap menurun mendekati 0%, menghasilkan estimasi yang semakin akurat dengan sampel yang lebih besar.





Estimasi Parameter dengan *Cluster Random Sampling*



Estimasi Parameter dengan *Cluster Random Sampling*

```
# Make cluster sampling from completed_data
cluster_sampling = completed_data.groupby(["Kabupaten_Kota", "Kecamatan",
                                           "Kelurahan"]).apply(lambda x: x.sample(frac=.1, replace=False))
cluster_sampling = cluster_sampling.reset_index(drop=True)
cluster_sampling
```

	Provinsi	Kabupaten_Kota	Kecamatan	Kelurahan	nomor_tps	suara1	suara2	golput
0	Kalimantan Tengah	Barito Selatan	Dusun Hilir	Mangkatir	5	129	17	10
1	Kalimantan Tengah	Barito Selatan	Dusun Hilir	Teluk Timbau	1	336	37	20
2	Kalimantan Tengah	Barito Selatan	Dusun Selatan	Baru	1	291	106	194
3	Kalimantan Tengah	Barito Selatan	Dusun Selatan	Buntok Kota	5	121	95	28
4	Kalimantan Tengah	Barito Selatan	Dusun Selatan	Danau Sadar	1	80	81	62
...
645	Kalimantan Tengah	Sukamara	Sukamara	Mendawai	1	76	103	33
646	Kalimantan Tengah	Sukamara	Sukamara	Mendawai	3	485	86	11
647	Kalimantan Tengah	Sukamara	Sukamara	Padang	6	26	44	42
648	Kalimantan Tengah	Sukamara	Sukamara	Petarikan	2	3	29	6
649	Kalimantan Tengah	Sukamara	Sukamara	Pudu	1	279	151	79

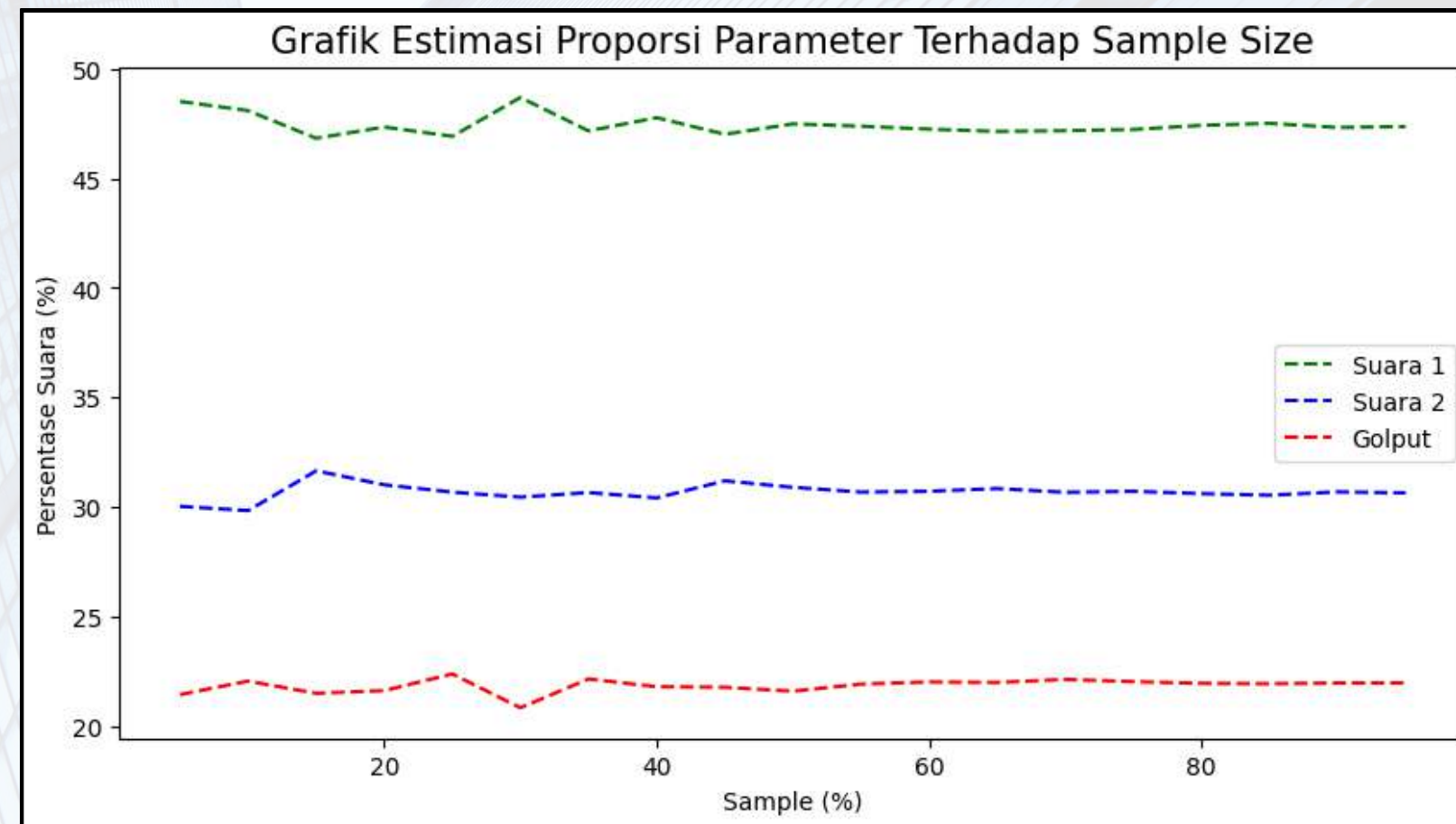
650 rows x 8 columns

Dalam tahap ini, akan dilakukan estimasi parameter dengan menerapkan metode *cluster random sampling*. Dataset akan diorganisir berdasarkan hierarki administratif, yaitu kabupaten/kota, kecamatan, kelurahan, dan nomor TPS. Sampel dataset yang telah dibentuk sebelumnya melalui *generate random* akan digunakan untuk membangun sampel dengan menggunakan pendekatan *cluster random sampling*.

Estimasi Parameter dengan *Cluster Random Sampling*

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

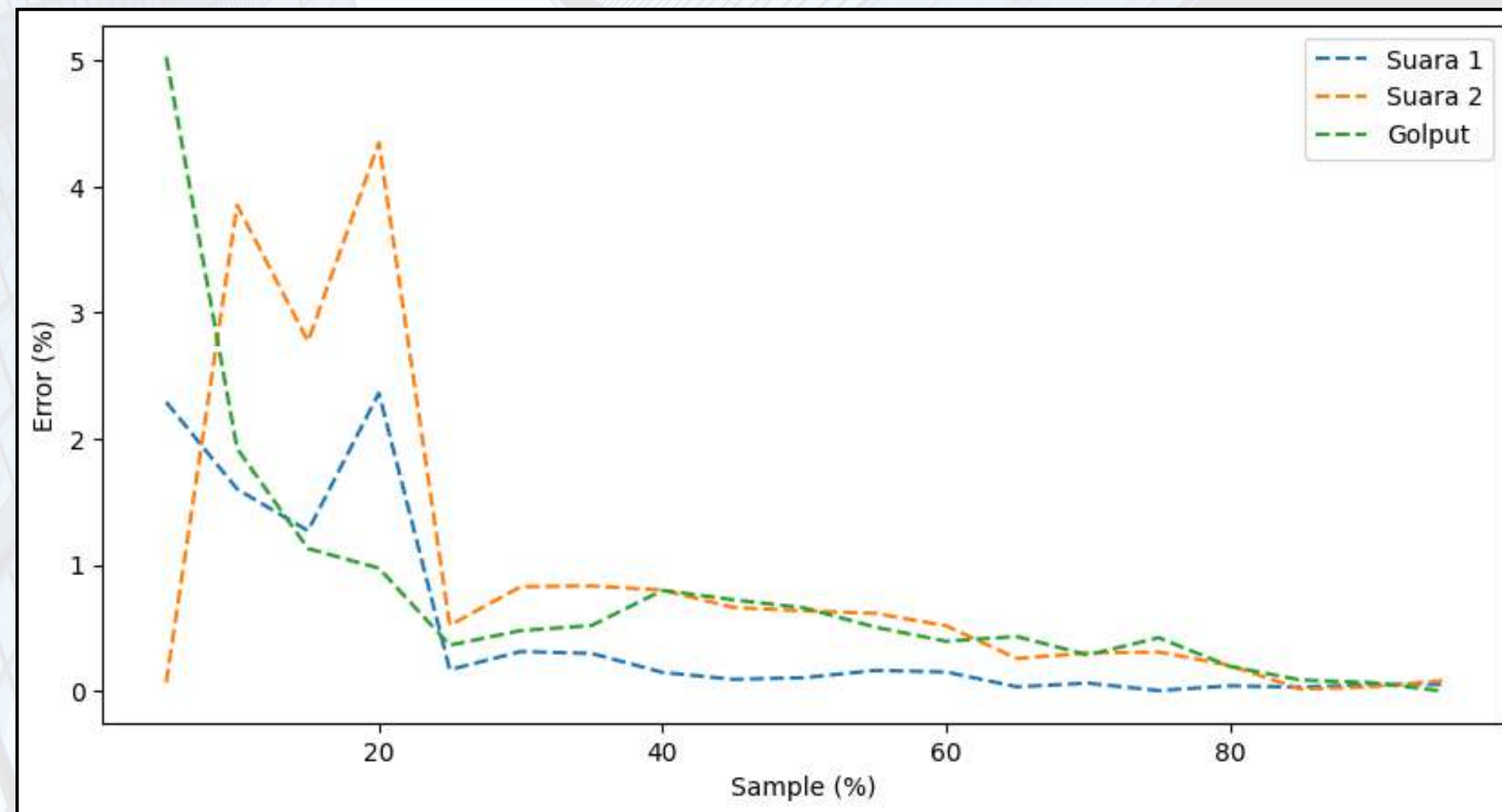
	Sample (%)	suara1	suara2	golput
0	5	48.506094	30.039909	21.453996
1	10	48.084786	29.846093	22.069121
2	15	46.825082	31.659573	21.515345
3	20	47.338498	31.022953	21.638550
4	25	46.915196	30.688365	22.396439
5	30	48.684812	30.459443	20.855745
6	35	47.161514	30.669940	22.168546
7	40	47.763297	30.423415	21.813288
8	45	47.007422	31.202723	21.789855
9	50	47.480601	30.907824	21.611575
10	55	47.374865	30.690765	21.934371
11	60	47.240782	30.726895	22.032322
12	65	47.139067	30.851937	22.008996
13	70	47.172418	30.680767	22.146815
14	75	47.226826	30.726061	22.047113
15	80	47.412324	30.620803	21.966873
16	85	47.504877	30.547027	21.948096
17	90	47.317129	30.700535	21.982335
18	95	47.357502	30.654770	21.987727



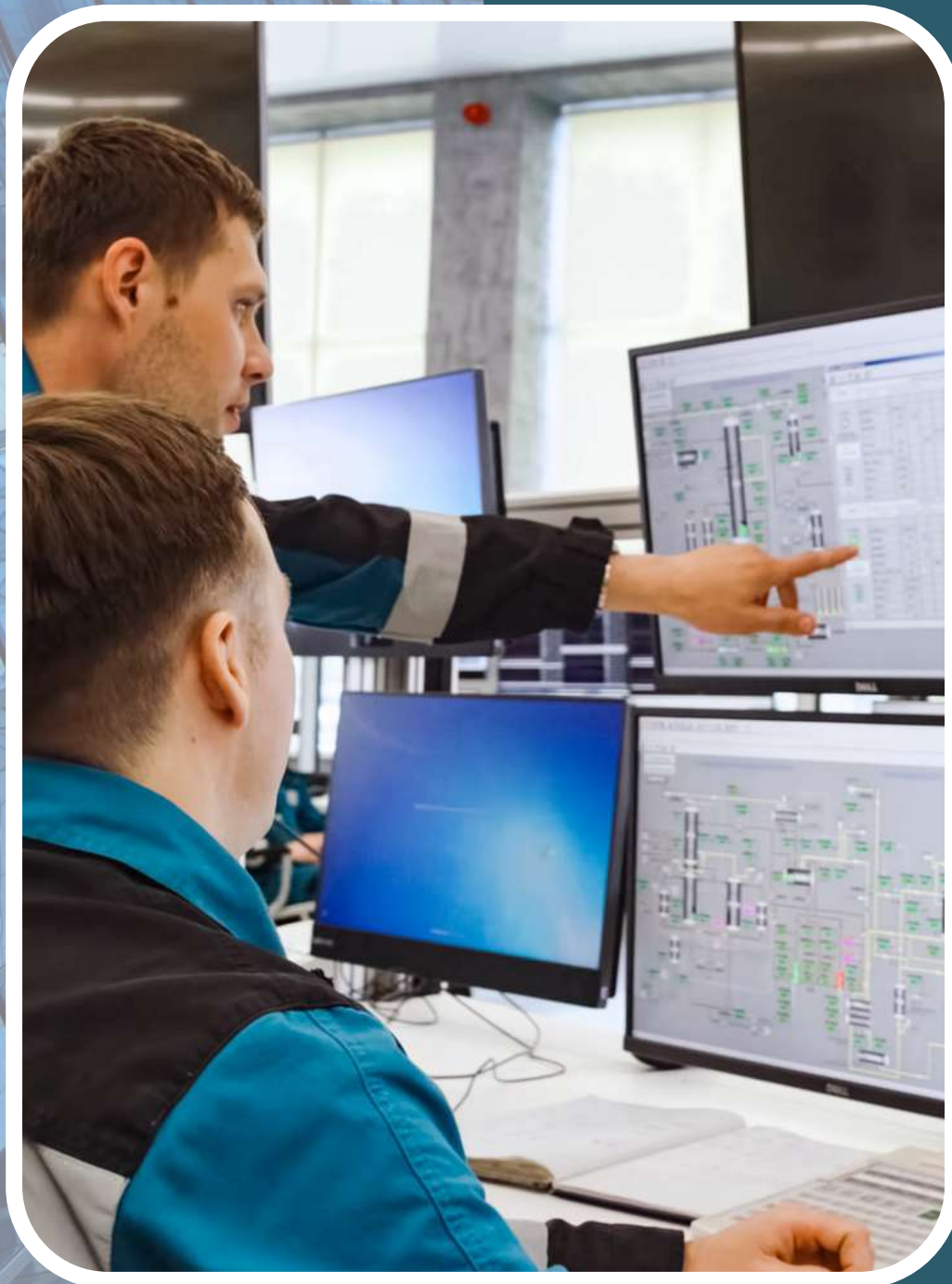
Tabel yang disajikan menunjukkan peningkatan ukuran sampel menghasilkan estimasi yang lebih stabil dan mendekati nilai sebenarnya. Analisis visual dari grafik estimasi parameter terhadap ukuran sampel menunjukkan tingkat homogenitas yang signifikan dalam estimasi proporsi pada setiap data sampel yang diambil dengan metode *cluster random sampling*. Hal ini dipengaruhi oleh penggunaan distribusi *Dirichlet* dalam pembentukan dan distribusi data suara dari setiap TPS. Hasil visualisasi menunjukkan bahwa persentase tertinggi adalah untuk suara1, diikuti oleh suara2, dan persentase paling rendah adalah untuk golput atau suara tidak sah. Terdapat perbedaan sekitar 5% antara estimasi suara2 dan golput dalam satu pengambilan data. Sementara itu, perbedaan antara estimasi suara1 dengan suara2 dan golput relatif lebih besar."

Melakukan Evaluasi Terhadap Tingkat Kesalahan (*Error*) dari Estimasi Metode *Cluster Random Sampling*

=====Error Estimasi Proporsi Parameter=====				
	Sample (%)	Error Suara 1	Error Suara 2	Error Golput
0	5	2.291150	0.066525	5.027619
1	10	1.600838	3.851202	1.921614
2	15	1.270475	2.772415	1.129054
3	20	2.361638	4.346706	0.973811
4	25	0.168592	0.521304	0.363713
5	30	0.313165	0.827398	0.479098
6	35	0.298083	0.833120	0.519562
7	40	0.147757	0.800401	0.797726
8	45	0.093033	0.662669	0.723561
9	50	0.105575	0.635763	0.659031
10	55	0.163485	0.616021	0.506775
11	60	0.151542	0.517156	0.394653
12	65	0.033574	0.258122	0.432207
13	70	0.062402	0.301584	0.286084
14	75	0.003973	0.309211	0.422567
15	80	0.041327	0.204333	0.195882
16	85	0.030836	0.015265	0.087701
17	90	0.052813	0.033782	0.066652
18	95	0.052847	0.081980	0.000475



Dari visualisasi grafik, terlihat bahwa metode *cluster random sampling* menghasilkan *error* maksimum mendekati 5% pada pengambilan pertama. Namun, *error* maksimum pada golput cenderung menurun seiring dengan peningkatan jumlah sampel, bahkan mencapai 0%. Penurunan *error* terjadi secara cepat ketika jumlah sampel melebihi 20%. Kedua garis yang mewakili suara dan golput semakin mendekati nilai 0% seiring dengan pertambahan jumlah sampel, menunjukkan tingkat akurasi yang lebih tinggi dengan sampel yang lebih besar.



Estimasi Parameter dengan *Multistage Random Sampling*



Estimasi Parameter dengan *Multistage Random Sampling*

```
# 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 completed_data
tps = []
for kel in kelurahan_list:
    tps.append(completed_data[completed_data["Kelurahan"] == kel].sample(n=3, replace=False))

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

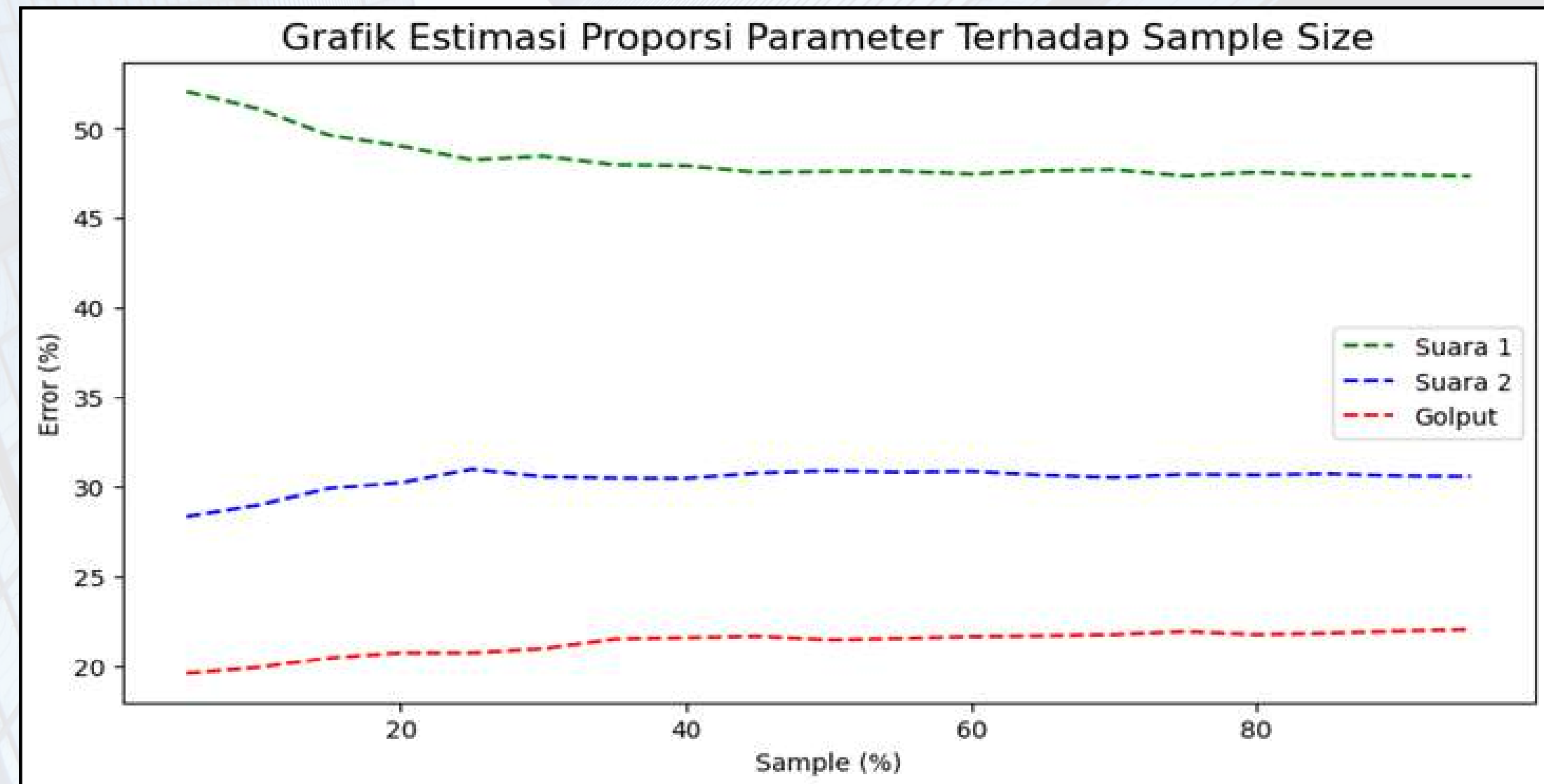
	Provinsi	Kabupaten_Kota	Kecamatan	Kelurahan	nomor_tps	suara1	suara2	golput
0	Kalimantan Tengah	Gunung Mas	Kahayan Hulu Utara	Tumbang Takaoi	4	9	2	19
1	Kalimantan Tengah	Gunung Mas	Kahayan Hulu Utara	Tumbang Takaoi	2	40	5	21
2	Kalimantan Tengah	Gunung Mas	Kahayan Hulu Utara	Tumbang Takaoi	3	123	2	42
3	Kalimantan Tengah	Kotawaringin Timur	Mentaya Hulu	Pantap	5	136	54	79
4	Kalimantan Tengah	Kotawaringin Timur	Mentaya Hulu	Pantap	4	6	38	62
...
145	Kalimantan Tengah	Kotawaringin Timur	Mentaya Hilir Selatan	Samuda Besar	1	39	117	18
146	Kalimantan Tengah	Kotawaringin Timur	Mentaya Hilir Selatan	Samuda Besar	3	276	51	20
147	Kalimantan Tengah	Gunung Mas	Mihing Raya	Rangan Tate	1	136	35	27
148	Kalimantan Tengah	Gunung Mas	Mihing Raya	Rangan Tate	4	14	6	6
149	Kalimantan Tengah	Gunung Mas	Mihing Raya	Rangan Tate	3	73	42	2

150 rows x 8 columns

Proses dimulai dengan menghapus baris yang memiliki nilai Provinsi dengan label TOTAL untuk membersihkan data. Selanjutnya, dilakukan pengambilan acak 50 kelurahan tanpa penggantian. Setiap kelurahan dipilih secara acak untuk menyusun daftar 3 Tempat Pemungutan Suara (TPS) dari setiap kelurahan. TPS yang terpilih dari setiap kelurahan digabungkan menjadi *dataframe* tunggal bernama ***cluster_sampling***. Indeks data direset untuk memastikan urutan yang benar. Hasilnya adalah *dataframe* ***cluster_sampling*** berisi 3 TPS secara acak dari setiap kelurahan melalui *sampling* acak non-ganda.

Estimasi Parameter dengan *Multistage Random Sampling*

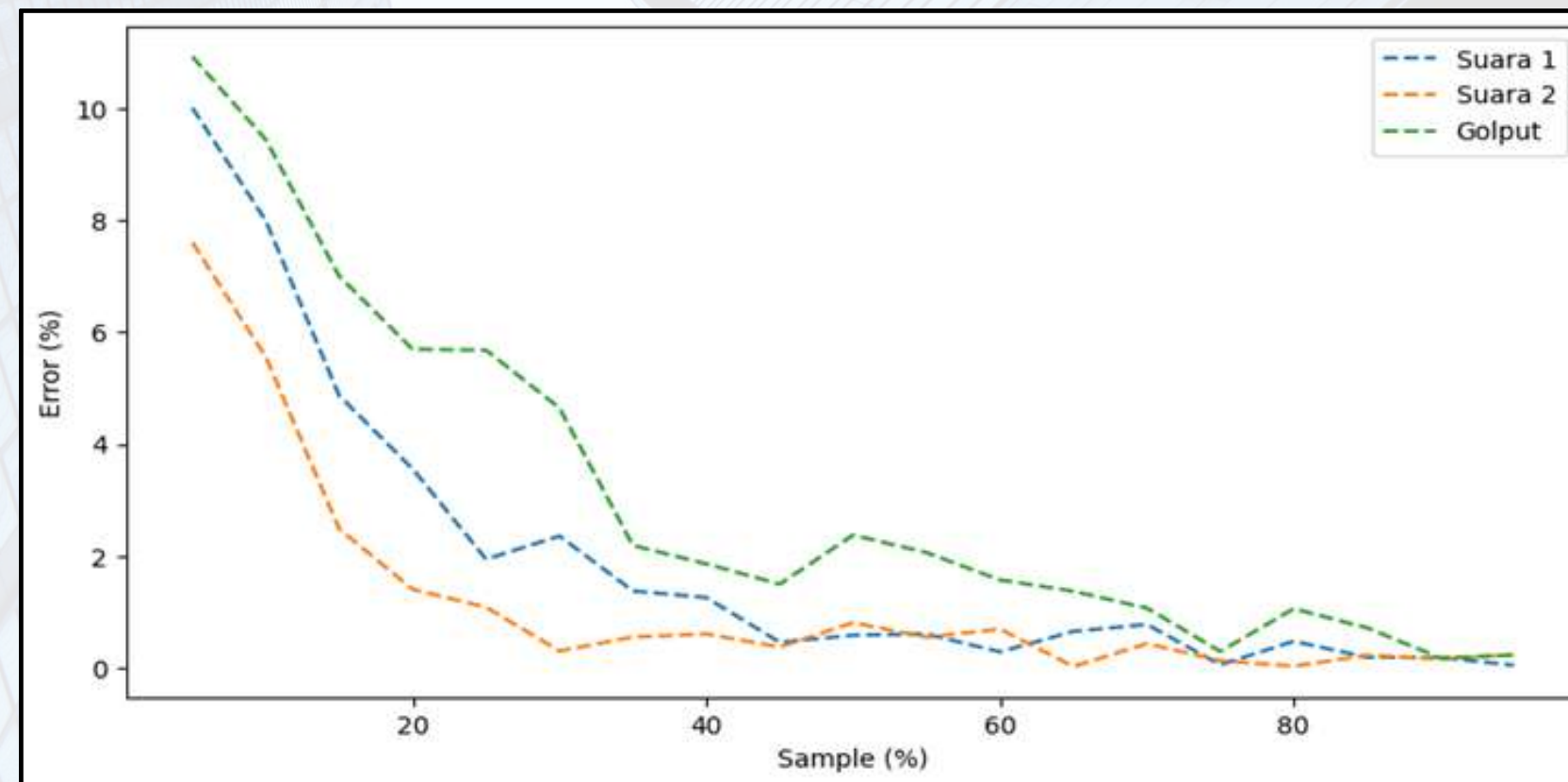
=====Estimasi Proporsi Parameter=====				
	Sample (%)	suara1	suara2	golput
0	5	52.060874	28.339548	19.599579
1	10	51.105883	28.967808	19.926308
2	15	49.624567	29.911574	20.463859
3	20	49.005549	30.243030	20.751421
4	25	48.239994	31.004915	20.755092
5	30	48.436503	30.580938	20.982559
6	35	47.971584	30.504715	21.523701
7	40	47.915965	30.487893	21.596142
8	45	47.537287	30.786737	21.675976
9	50	47.596682	30.920612	21.482706
10	55	47.607390	30.840494	21.552116
11	60	47.455928	30.884571	21.659501
12	65	47.629765	30.666276	21.703960
13	70	47.688589	30.540976	21.770435
14	75	47.347071	30.712169	21.940760
15	80	47.545908	30.681553	21.772539
16	85	47.410317	30.741462	21.848221
17	90	47.408181	30.624889	21.966930
18	95	47.342470	30.602440	22.055090



Hasil *plot* menunjukkan bahwa dengan peningkatan proporsi data sampel, terlihat tren peningkatan persentase suara2 dan golput, sementara persentase suara1 cenderung menurun. Estimasi suara1 juga secara konsisten lebih tinggi daripada suara2 dan golput.

Melakukan Evaluasi Terhadap Tingkat Kesalahan (*Error*) dari Estimasi Metode *Multistage Random Sampling*

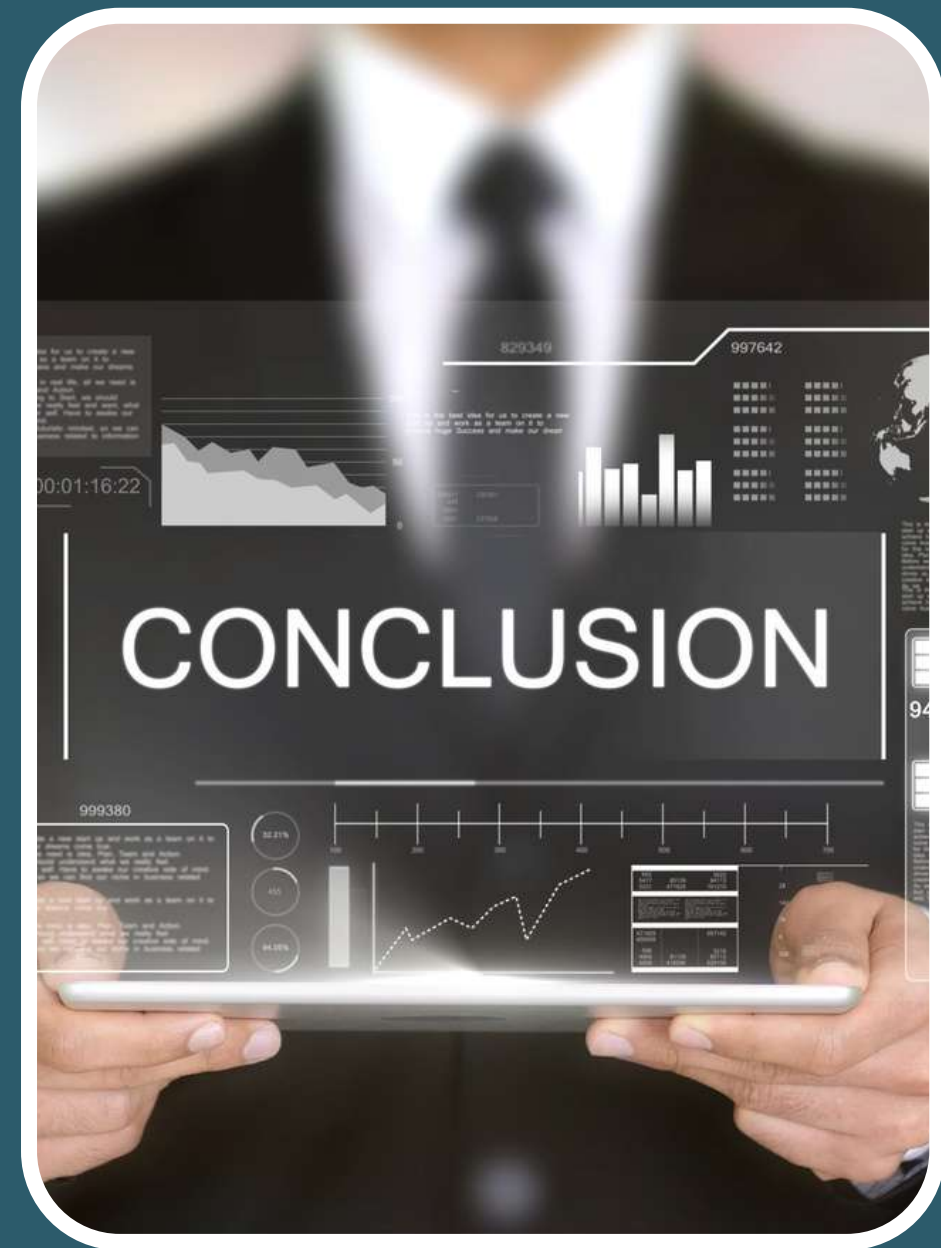
=====Error Estimasi Proporsi Parameter=====				
	Sample (%)	Error Suara 1	Error Suara 2	Error Golput
0	5	10.014079	7.607681	10.930981
1	10	7.996010	5.559431	9.446178
2	15	4.865720	2.482576	7.003313
3	20	3.557622	1.401966	5.696506
4	25	1.939865	1.081922	5.679826
5	30	2.355125	0.300322	4.646115
6	35	1.372667	0.548822	2.186933
7	40	1.255134	0.603667	1.857727
8	45	0.454919	0.370623	1.494929
9	50	0.580431	0.807081	2.373232
10	55	0.603059	0.545880	2.057800
11	60	0.282992	0.689579	1.569795
12	65	0.650340	0.022104	1.367757
13	70	0.774647	0.430606	1.065662
14	75	0.052957	0.127516	0.291633
15	80	0.473137	0.027702	1.056103
16	85	0.186608	0.223018	0.712172
17	90	0.182094	0.157033	0.172706
18	95	0.043234	0.230221	0.227933



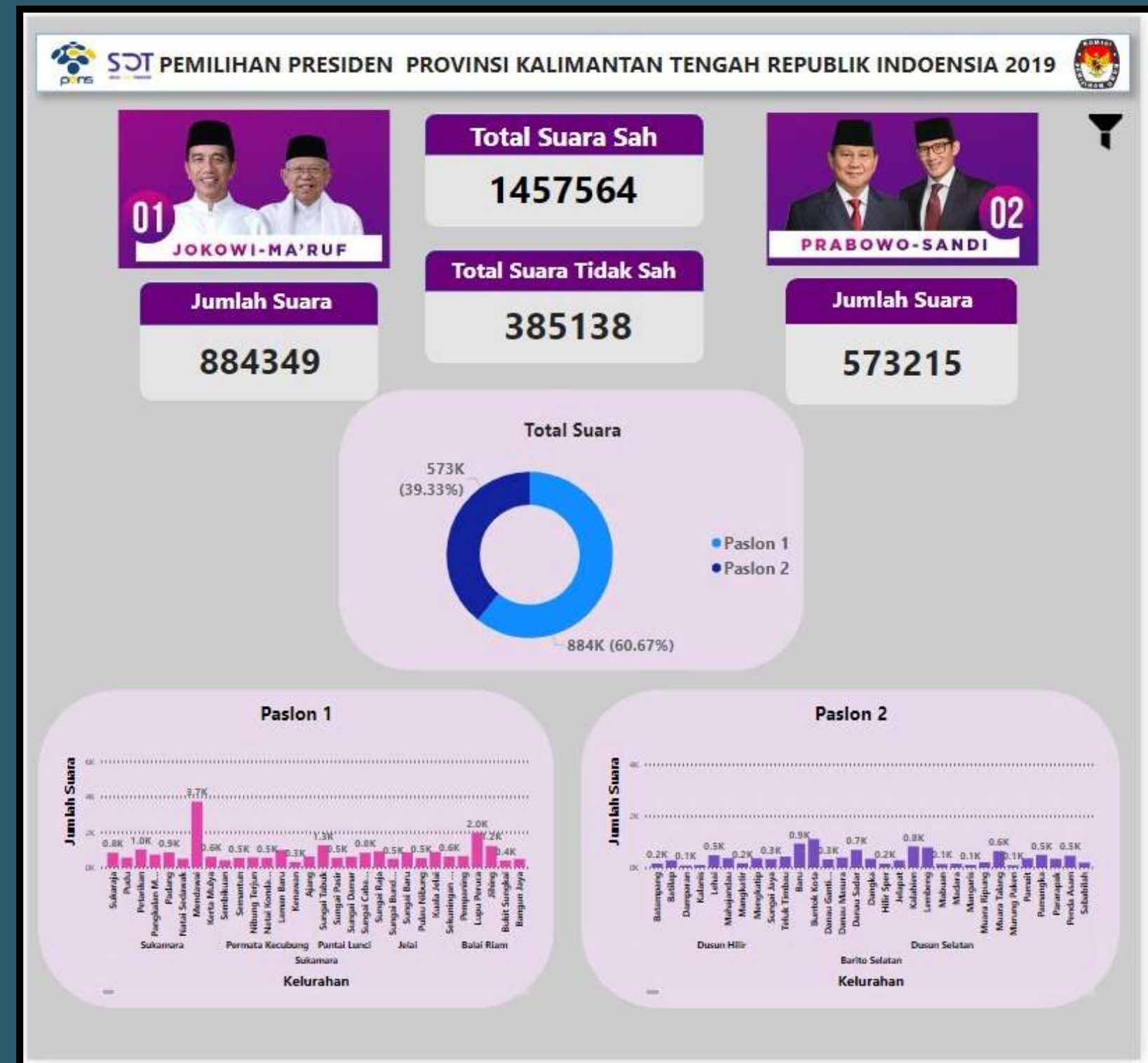
Data ini memberikan wawasan mengenai performa metode *sampling* dalam mendekati parameter populasi. *Plot error* dari pengujian dengan metode *multistage random sampling* menunjukkan nilai *error* maksimum sekitar 10% dalam satu pengambilan sampel. *Error* cenderung menurun dengan peningkatan proporsi sampel, hingga mendekati 0 seiring pertambahan sampel, menunjukkan bahwa ukuran sampel mempengaruhi akurasi estimasi.

KESIMPULAN

Penelitian ini membahas tiga metode *sampling* untuk *quick count* dalam pemilihan umum: *simple random sampling*, *cluster random sampling*, dan *multistage random sampling*. Ukuran sampel berpengaruh signifikan terhadap kualitas estimasi parameter, semakin besar sampel, estimasi semakin baik, dan tingkat *error* estimasi menurun seiring dengan peningkatan ukuran sampel. Evaluasi menunjukkan bahwa *simple random sampling* efektif mendekati nilai sebenarnya dari populasi dengan tingkat *error* rendah. Analisis distribusi dukungan pemilih di provinsi Kalimantan Tengah tahun 2019 menunjukkan dominasi suara1, memberikan wawasan yang penting. Kesimpulannya, *simple random sampling* dengan sampel yang cukup besar adalah metode yang efektif, memberikan estimasi akurat dan wawasan yang relevan dalam konteks distribusi dukungan pemilih pada pemilihan presiden tahun 2019 di provinsi Kalimantan Tengah.

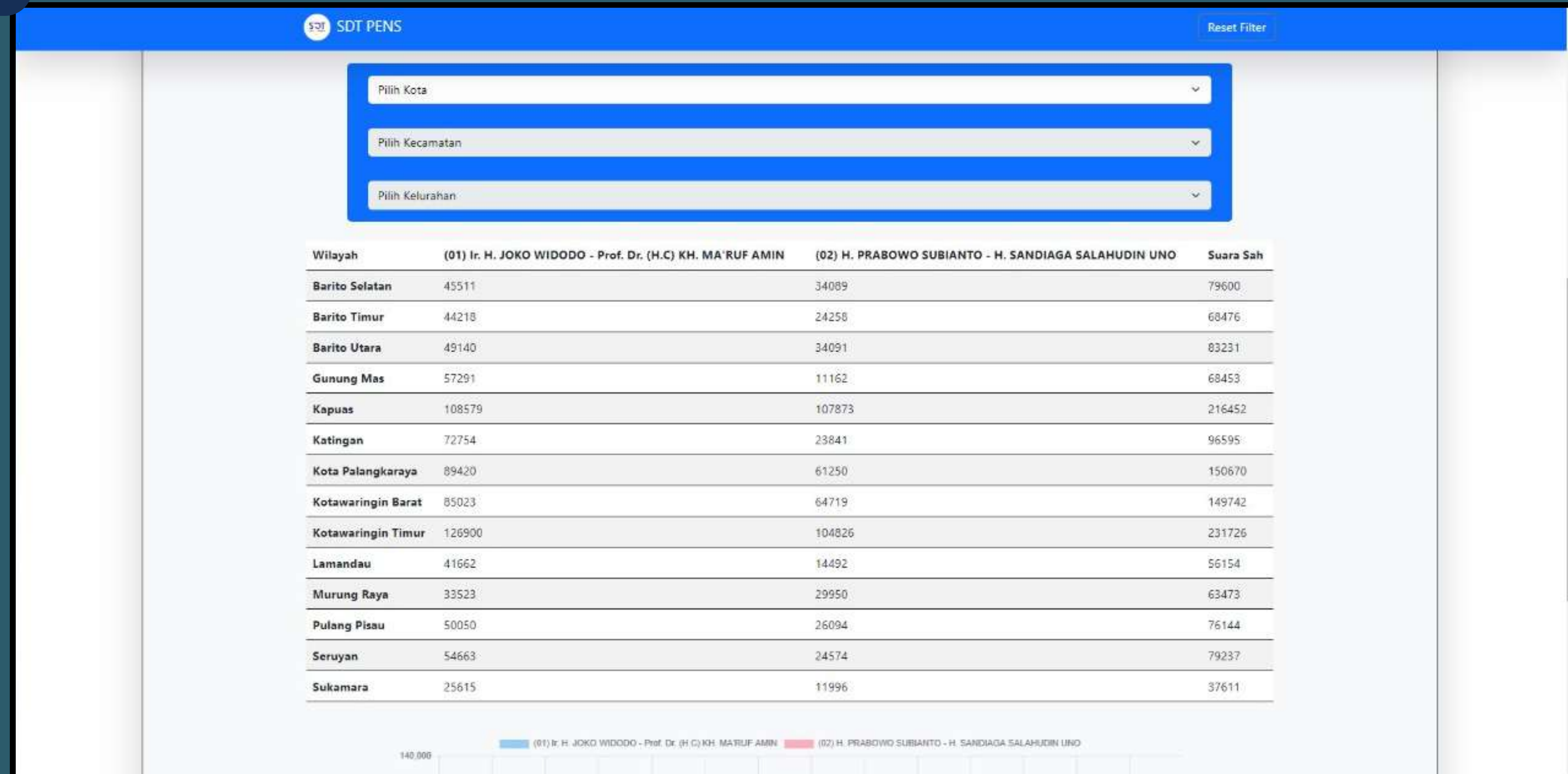


Lampiran Dashboard



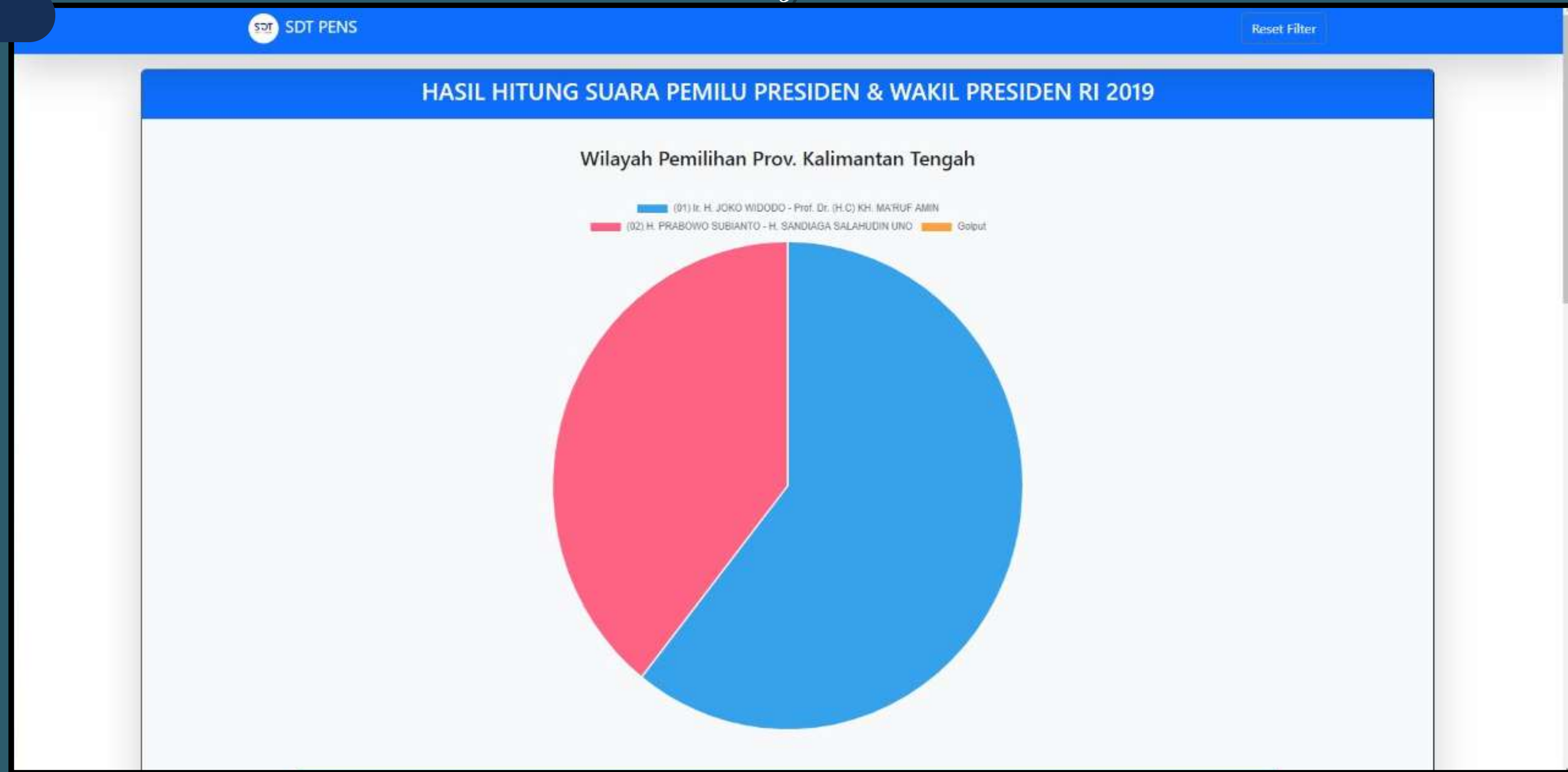
Dashboard dengan Software Visualisasi PowerBI

Lampiran Dashboard



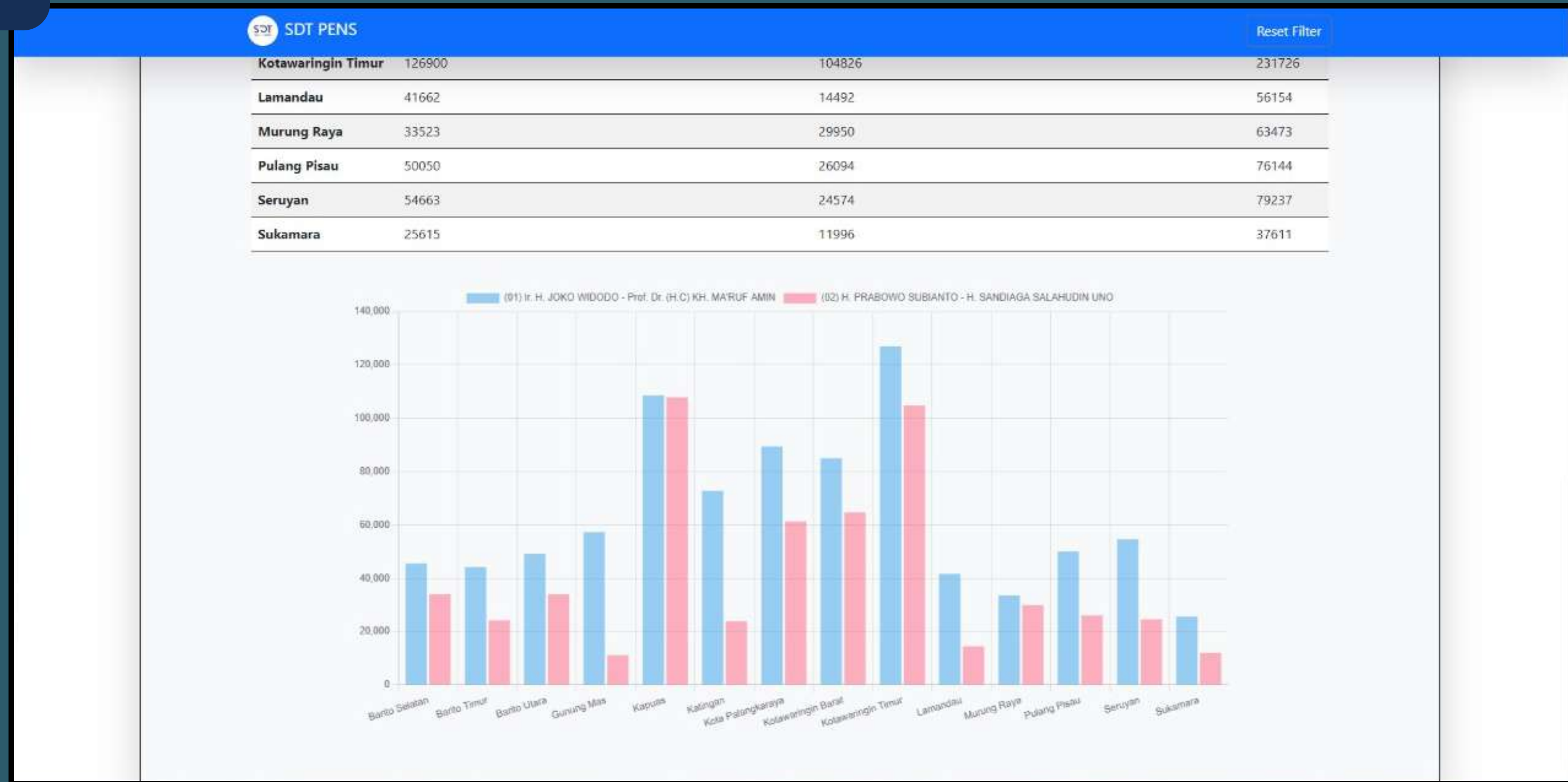
Dashboard dengan Berbasis Web: Tampilan Tabel Suara

Lampiran Dashboard



Dashboard dengan Berbasis Web: Tampilan Pie Chart

Lampiran Dashboard



Dashboard dengan Berbasis Web: Tampilan Bar Chart Suara Tiap Kabupaten



thank
you

.terima kasih

