

PROJECT UAS STATISTIKA PENGENDALIAN MUTU

**Analisis Garis Kemiskinan Menurut
Kabupaten/Kota (Rupiah/kapita/bulan) pada Tahun
2019-2023 dengan Menggunakan Metode Bagan
Kendali Variabel**



Disusun oleh :

Nadia Sukesu Sianipar

(2106700776)

Statistika Pengendalian Mutu

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN

UNIVERSITAS INDONESIA

TAHUN AJARAN 2023/2024

Bagian 1. Pendahuluan

Indonesia sebagai negara berkembang harus memiliki niat dan itikad yang kuat untuk belajar dari negara-negara maju. Indonesia masih harus mengejar ketertinggalan dari negara lain dalam berbagai hal dengan cara melakukan pembangunan yang berkelanjutan. Pembangunan dilaksanakan untuk dapat mewujudkan kemakmuran dan kesejahteraan masyarakat. Berbagai strategi pembangunan ekonomi dilakukan oleh pemerintah untuk berorientasi pada peningkatan pertumbuhan ekonomi dan pemerataan ekonomi yang berkeadilan yaitu dengan menurunkan jumlah penduduk miskin, namun dalam realitasnya masih terdapat diskrepansi antara harapan dan kenyataan di lapangan berupa kemiskinan. Berdasarkan *World Bank*, Indonesia memiliki tingkat kemiskinan yang tertinggi nomor 2 setelah Laos. Tingkat kemiskinan di Indonesia mencapai 41,7 persen pada tahun 2016. Hal ini menggambarkan bahwa, 41,7 persen dari total penduduk Indonesia memiliki pendapatan kurang dari standar rasio tingkat kemiskinan yang telah ditetapkan oleh *World Bank* atau masih berada di bawah garis kemiskinan

Pada era globalisasi yang ditandai dengan pesatnya perkembangan teknologi dan ekonomi, analisis statistik memiliki peran yang semakin penting dalam mendukung pengambilan keputusan yang tepat, terutama dalam konteks pembangunan sosial dan ekonomi yaitu analisis garis kemiskinan. Analisis ini menjadi indikator kunci dalam mengevaluasi tingkat kemiskinan suatu daerah di Indonesia. Garis kemiskinan menunjukkan batas minimum pengeluaran yang diperlukan oleh individu atau rumah tangga untuk memenuhi kebutuhan dasar seperti makanan, pakaian, dan tempat tinggal.

Di Indonesia, Badan Pusat Statistik (BPS) secara rutin merilis data garis kemiskinan yang memberikan gambaran tentang tingkat kemiskinan di berbagai wilayah. Namun, lebih dari sekadar menyajikan data, penting untuk menganalisis tren dan pola perubahan kemiskinan dari waktu ke waktu untuk mendapatkan pemahaman yang lebih mendalam tentang dinamika kemiskinan di Indonesia.

Analisis kemiskinan sangat penting dalam konteks pembangunan nasional. Kemiskinan bukan hanya masalah ekonomi, tetapi juga masalah sosial yang kompleks, yang berkaitan dengan akses terhadap pendidikan, kesehatan, pekerjaan layak, dan keadilan sosial. Oleh karena itu, pemahaman yang baik tentang kemiskinan diperlukan untuk merancang kebijakan yang efektif dalam mengurangi ketimpangan sosial dan meningkatkan kesejahteraan masyarakat.

Dalam penelitian ini, penulis akan menggunakan metode Bagan Kendali Variabel, khususnya Bagan Kendali X-R, untuk menganalisis garis kemiskinan menurut kabupaten/kota di Indonesia pada periode tahun 2019-2023. Bagan Kendali X-R adalah alat analisis statistik yang berguna dalam memantau dan mengevaluasi perubahan dalam suatu proses atau fenomena dari waktu ke waktu. Dalam konteks ini, Bagan Kendali X-R akan membantu penulis dalam mengidentifikasi perubahan signifikan dalam tingkat kemiskinan, serta mengidentifikasi kabupaten/kota yang memerlukan perhatian khusus dalam upaya penanggulangan kemiskinan.

Data pada penelitian ini diambil dari Badan Pusat Statistik yang dapat diakses melalui website (<https://www.bps.go.id/id/statistics-table/2/NjI0IzI=/garis-kemiskinan-menurut-kabupaten-kota.html>).

Data penelitian ini berukuran 547 subgroup dengan jumlah pengukuran 6 kolom data yaitu Wilayah, Tahun 2019, 2020, 2021, 2022, dan 2023. Tipe skala data penelitian ini adalah data kontinu.

Angka-angka pada data menggambarkan garis kemiskinan di berbagai wilayah di Indonesia dari tahun 2019 hingga 2023. Berikut adalah penjelasan dari setiap kolom dalam data tersebut:

- Wilayah: Nama wilayah atau kabupaten/kota di Indonesia
- Tahun 2019: Garis kemiskinan pada tahun 2019, diukur dalam Rupiah per kapita per bulan. Garis kemiskinan menunjukkan jumlah minimum pengeluaran yang diperlukan oleh individu atau rumah tangga untuk memenuhi kebutuhan dasar seperti makanan, pakaian, dan tempat tinggal.
- Tahun 2020: Garis kemiskinan pada tahun 2020, diukur dalam Rupiah per kapita per bulan.
- Tahun 2021: Garis kemiskinan pada tahun 2021, diukur dalam Rupiah per kapita per bulan.
- Tahun 2022: Garis kemiskinan pada tahun 2022, diukur dalam Rupiah per kapita per bulan.
- Tahun 2023: Garis kemiskinan pada tahun 2023, diukur dalam Rupiah per kapita per bulan.

Setiap angka pada tabel mewakili jumlah Rupiah yang dibutuhkan oleh seorang individu di wilayah tertentu untuk memenuhi kebutuhan dasar mereka dalam satu bulan pada tahun tersebut. Sebagai contoh

- ACEH 2019: 486,935: Ini berarti di wilayah Aceh, seorang individu memerlukan setidaknya 486,935 Rupiah per bulan untuk memenuhi kebutuhan dasar mereka pada tahun 2019.

Bagian 2. *Pre-processing* dan analisis deskriptif untuk visualisasi data

Preprocessing adalah tahap penting dalam analisis data dan machine learning yang melibatkan berbagai teknik untuk mempersiapkan data mentah sehingga siap digunakan dalam analisis atau pemodelan lebih lanjut. Proses ini bertujuan untuk membersihkan, merapikan, dan mengatur data agar dapat menghasilkan hasil yang lebih akurat dan dapat diandalkan. Berikut adalah langkah-langkah *pre-processing* yang dilakukan pada penelitian ini :

1. *Import Modul*

- **import numpy as np:** NumPy adalah modul dalam Python yang digunakan untuk melakukan operasi matematika pada array dan matriks. Penggunaan np sebagai alias memudahkan dalam memanggil fungsi-fungsi dari NumPy.
- **import os:** Modul os menyediakan fungsi-fungsi yang memungkinkan penulis berinteraksi dengan sistem operasi, seperti mengakses file dan direktori.

- **import pandas as pd:** Pandas adalah modul yang digunakan untuk manipulasi dan analisis data. Biasanya, penulis menggunakan DataFrame dari Pandas untuk memuat dan memanipulasi data tabular.
- **import matplotlib.pyplot as plt:** Matplotlib adalah modul yang digunakan untuk membuat visualisasi data, seperti plot grafik. Dengan menggunakan plt sebagai alias, penulis dapat dengan mudah memanggil fungsi-fungsi dari Matplotlib.

2. Import Data

Pada langkah ini, penulis menggunakan Google Colab untuk mengimpor data dari Google Drive. Ini adalah langkah awal dalam memuat data ke dalam lingkungan kerja di Google Colab.

- **from google.colab import drive**

Dalam baris pertama, penulis mengimpor modul `drive` dari pustaka `google.colab`. Modul ini digunakan untuk mengakses Google Drive dari Google Colab.

- **drive.mount('/content/drive')**

Baris kedua ini adalah perintah untuk melakukan mount atau menghubungkan Google Drive penulis ke lingkungan kerja Google Colab. `/content/drive` adalah direktori di Google Colab tempat Google Drive akan di-mount.

- **data = '/content/drive/MyDrive/Stat QC/Garis Kemiskinan Menurut Kabupaten_Kota 2019-2023.xlsx'**

Pada baris ini, penulis menentukan lokasi file Excel yang ingin penulis baca dan masukkan ke dalam variabel `data`. Penulis menentukan alamat file dari Google Drive penulis.

- **ukuran = '/content/drive/MyDrive/Stat QC/x-bar-r.xlsx'**

Sama seperti baris sebelumnya, pada baris ini penulis menentukan lokasi file Excel yang ingin penulis baca dan masukkan ke dalam variabel `ukuran`.

- **df = pd.read_excel(data)**

Dalam baris ini, penulis menggunakan fungsi `read_excel()` dari Pandas untuk membaca file Excel yang telah penulis tentukan sebelumnya dalam variabel `data`. Data yang dibaca kemudian dimuat ke dalam DataFrame Pandas dengan nama `df`.

- **df1 = pd.read_excel(ukuran)**

Langkah yang sama juga dilakukan di baris ini, namun data yang dibaca berasal dari file yang telah ditentukan sebelumnya dalam variabel `ukuran`. Data kemudian dimuat ke dalam DataFrame Pandas dengan nama `df1`.

3. Menampilkan Data

Dengan menggunakan method head() pada DataFrame df, penulis dapat menampilkan lima baris pertama dari DataFrame tersebut. Method head() digunakan untuk melihat preview atau

gambaran singkat dari data yang kita miliki. Ini membantu kita untuk memeriksa apakah data telah dimuat dengan benar dan memahami struktur data tersebut.

```
# Menampilkan beberapa baris pertama dari dataframe
print(df.head())
```

	Wilayah	Tahun 2019	Tahun 2020	Tahun 2021	Tahun 2022	Tahun 2023
0	ACEH	486935	522126	541109	579227	627534
1	Simeulue	404739	444754	458896	493303	538693
2	Aceh Singkil	450217	473983	487249	518951	568691
3	Aceh Selatan	369107	405786	418689	446224	494565
4	Aceh Tenggara	357015	392493	404725	430825	471301

```
[54] # Menampilkan beberapa baris pertama dari tabel ukuran yang akan digunakan
print(df1.head())
```

	Sample_size	A2	d2	D3	D4
0	2	1.880	1.128	0.0	3.267
1	3	1.023	1.693	0.0	2.574
2	4	0.729	2.059	0.0	2.282
3	5	0.577	2.326	0.0	2.114
4	6	0.483	2.534	0.0	2.004

4. Analisis Deskriptif

```
df.describe()
```

	Tahun 2019	Tahun 2020	Tahun 2021	Tahun 2022	Tahun 2023
count	548.000000	5.480000e+02	5.480000e+02	5.480000e+02	5.480000e+02
mean	416462.686131	4.447421e+05	4.638927e+05	4.937062e+05	5.351946e+05
std	101159.003128	1.068246e+05	1.109558e+05	1.166011e+05	1.235978e+05
min	235654.000000	2.481840e+05	2.538770e+05	2.646660e+05	2.902030e+05
25%	342226.000000	3.656400e+05	3.806308e+05	4.077020e+05	4.424072e+05
50%	391578.000000	4.225370e+05	4.406395e+05	4.725090e+05	5.134520e+05
75%	465315.750000	4.981728e+05	5.194238e+05	5.528855e+05	6.004872e+05
max	992610.000000	1.021759e+06	1.051297e+06	1.099019e+06	1.147265e+06

- **Tahun 2019, Tahun 2020, Tahun 2021, Tahun 2022, Tahun 2023:** Setiap kolom ini mewakili tahun yang diamati dalam data. Data statistik deskriptif dihitung untuk setiap tahun secara terpisah, memungkinkan penulis untuk melihat bagaimana distribusi data berubah dari tahun ke tahun.
- **Count (Jumlah):** Jumlah data yang diamati dalam setiap tahun adalah 548. Ini menunjukkan bahwa penulis memiliki 548 entri data untuk setiap tahun, sehingga penulis memiliki dataset yang cukup besar dan cukup representatif untuk melakukan analisis.
- **Mean (Rata-rata):** Nilai rata-rata dalam setiap tahun menunjukkan nilai tengah dari data. Rata-rata memberikan gambaran tentang nilai tengah atau pusat dari distribusi data. Dalam konteks ini, rata-rata menunjukkan nilai rata-rata kemiskinan per kapita per bulan dalam setiap tahun.

- **Std (Standar Deviasi):** Standar deviasi adalah ukuran seberapa jauh data tersebar dari rata-rata. Semakin tinggi standar deviasi, semakin tersebar data dari rata-rata. Dalam konteks ini, standar deviasi memberikan informasi tentang seberapa bervariasi tingkat kemiskinan per kapita per bulan dalam setiap tahun.
- **Min (Nilai Minimum):** Nilai minimum adalah nilai terendah dalam setiap tahun. Ini menunjukkan nilai terkecil yang diamati dalam dataset. Dalam hal ini, nilai minimum memberikan gambaran tentang tingkat minimum kemiskinan per kapita per bulan dalam setiap tahun.
- **25%, 50%, 75% (Persentil):** Persentil memberikan informasi tentang seberapa tersebar data di seluruh rentang nilainya. Misalnya, persentil ke-25 (25%) menunjukkan nilai di mana 25% dari data berada di bawahnya. Dalam konteks ini, persentil membantu penulis memahami distribusi data dengan lebih baik, termasuk seberapa banyak data yang terkonsentrasi di sekitar nilai rata-rata dan seberapa banyak data yang tersebar.
- **Max (Nilai Maksimum):** Nilai maksimum adalah nilai tertinggi dalam setiap tahun. Ini menunjukkan nilai tertinggi yang diamati dalam dataset. Dalam hal ini, nilai maksimum memberikan gambaran tentang tingkat maksimum kemiskinan per kapita per bulan dalam setiap tahun.

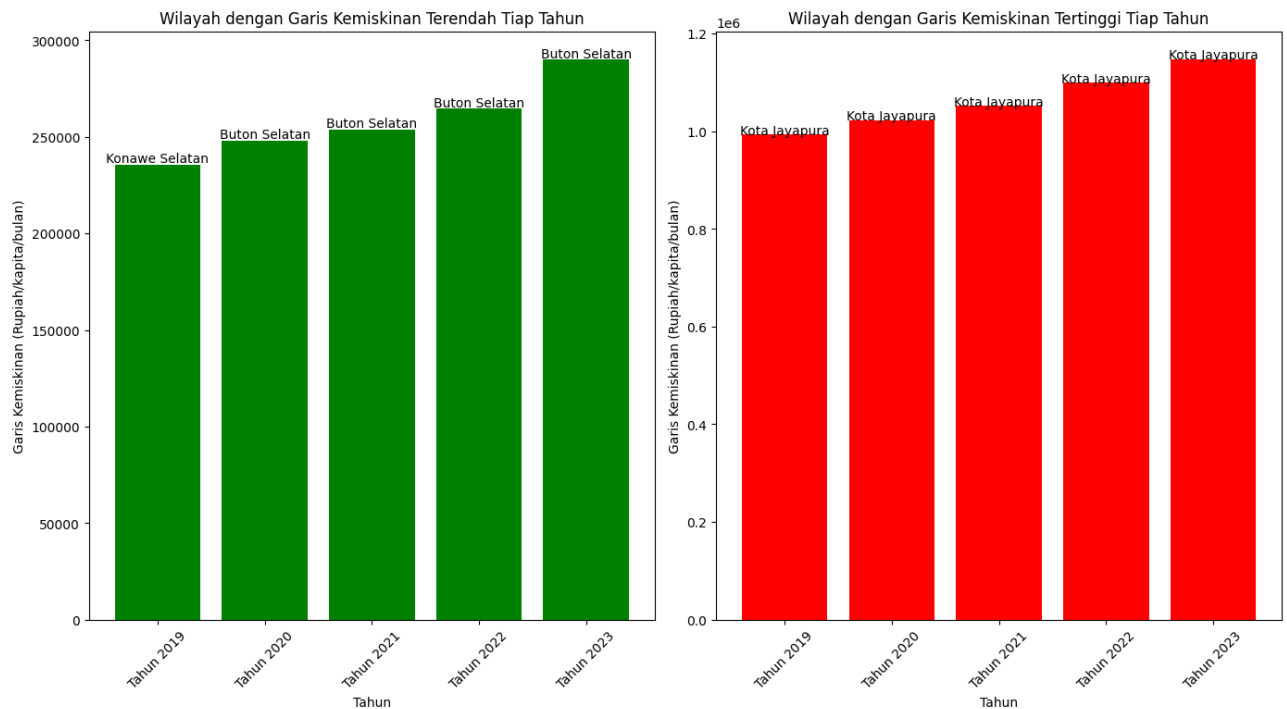
Dengan memahami setiap komponen dari tabel statistik deskriptif ini, penulis dapat memperoleh wawasan yang lebih dalam tentang distribusi dan karakteristik data dalam setiap tahun. Hal ini membantu dalam mengevaluasi tren, pola, serta variasi dalam tingkat kemiskinan per kapita per bulan dari tahun ke tahun.

5. Visualisasi

- Diagram Batang

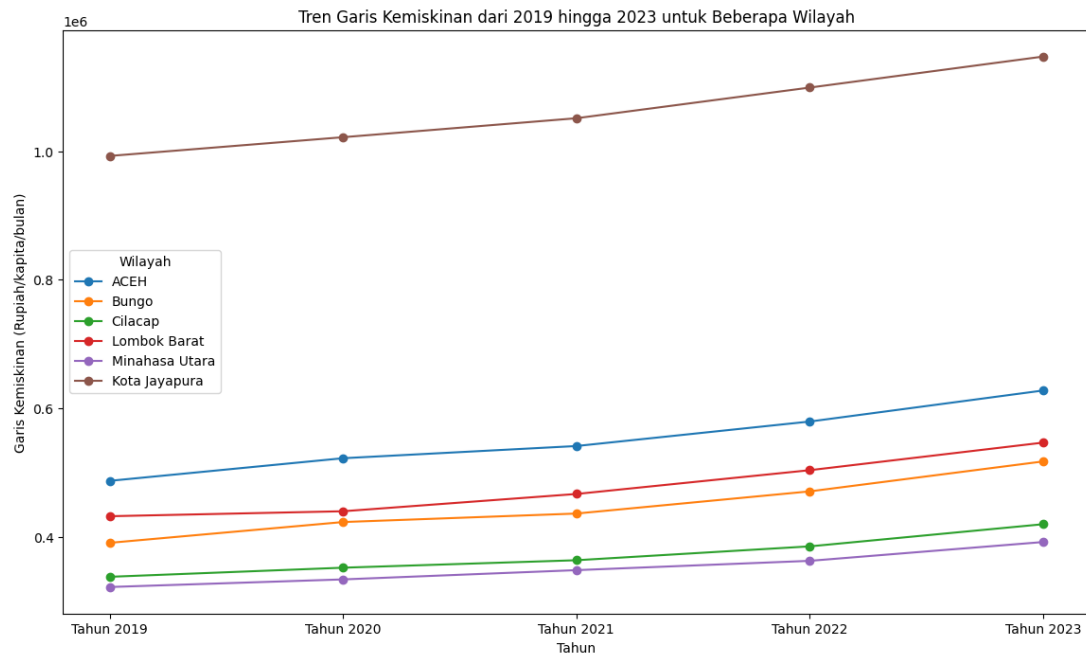
Diagram ini digunakan untuk menunjukkan wilayah dengan angka garis kemiskinan 5 terendah dan 5 tertinggi tiap tahun nya (2019 - 2023) di Indonesia. Dapat dilihat bahwa pada 5 tahun terakhir, Kota Jayapura sebagai kota dengan garis kemiskinan tertinggi secara berturut. Sebaliknya, kota Buton sebagai kota dengan garis kemiskinan terendah selama 4 Tahun terakhir.

	Year	Lowest	Highest	Lowest Value	Highest Value
0	Tahun 2019	Konawe Selatan	Kota Jayapura	235654	992610
1	Tahun 2020	Buton Selatan	Kota Jayapura	248184	1021759
2	Tahun 2021	Buton Selatan	Kota Jayapura	253877	1051297
3	Tahun 2022	Buton Selatan	Kota Jayapura	264666	1099019
4	Tahun 2023	Buton Selatan	Kota Jayapura	290203	1147265



- Line Plot

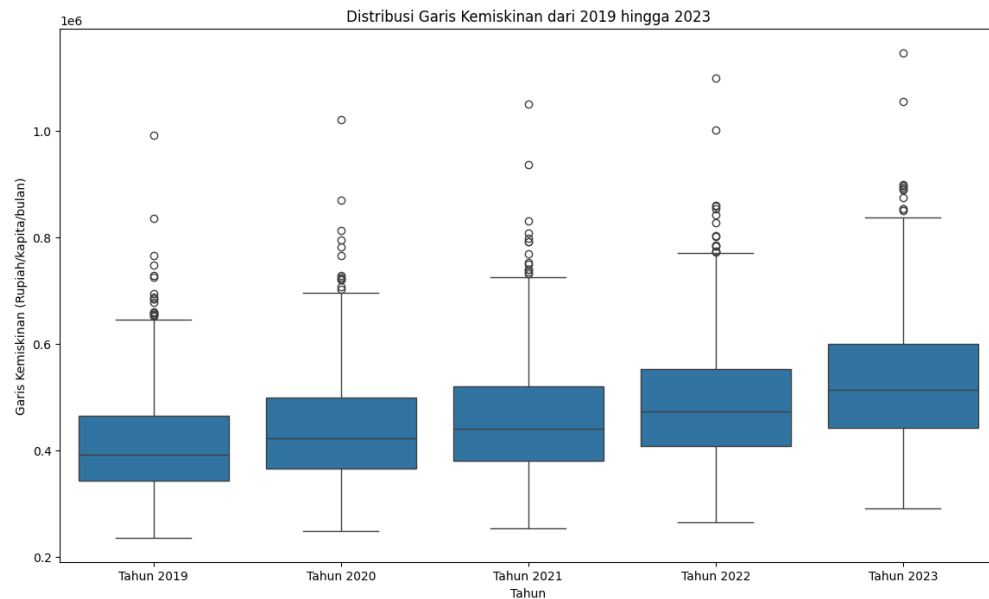
Menampilkan tren untuk setiap wilayah akan sulit dilakukan dalam satu plot, karena terdapat 548 wilayah pada data. Sebagai gantinya penulis memilih 5 wilayah secara acak untuk melihat tren garis kemiskinan. Kota/kabupaten yang dipilih adalah ACEH, Bungo, Cilacap, Lombok Barat, Minahasa Utara, Kota Jayapura. Dapat dilihat tren naik garis kemiskinan pada kelima kota/kabupaten ini.



- Box Plot

Box plot dapat membantu memahami distribusi garis kemiskinan di berbagai tahun. Dapat dilihat bahwa terdapat beberapa outlier. Outlier adalah nilai-nilai yang sangat

jauh dari sebagian besar data dan dapat mempengaruhi analisis statistik. Dalam penelitian ini, outlier tidak dihapus karena dibutuhkan untuk proses pengecekan menggunakan bagan kendali.



Bagian 3. Metode

Dalam penelitian mengenai analisis garis kemiskinan menurut kabupaten/kota pada periode tahun 2019-2023, pemilihan metode analisis data menjadi langkah krusial dalam memperoleh pemahaman yang mendalam tentang dinamika kemiskinan di Indonesia. Dalam konteks ini, metode Bagan Kendali Variabel, khususnya Peta Kendali X dan R, dipilih sebagai alat analisis utama. Alasan di balik pemilihan metode ini antara lain :

- Pertama, Peta Kendali X dan R dipilih karena kemampuannya dalam mendeteksi variasi dalam proses produksi atau fenomena lainnya. Dalam kasus ini, penulis menggunakan metode ini untuk memantau variasi dalam tingkat kemiskinan per kapita per bulan di setiap kabupaten/kota selama lima tahun terakhir. Kemampuan Peta Kendali X dan R untuk memberikan gambaran visual yang jelas tentang perubahan yang signifikan dalam tingkat kemiskinan sangat penting dalam memahami tren dan pola perubahan dari waktu ke waktu.
- Kedua, data yang bersifat kontinu dan terdiri dari sub group > 3. Data seperti itu akan sangat cocok dianalisis menggunakan peta kendali X dan R. Peta kendali X dan R digunakan untuk memantau proses yang mempunyai karakteristik berdimensi kontinu, sehingga disebut sebagai diagram kendali untuk data variabel
- Ketiga, kemudahan interpretasi dan implementasi Peta Kendali X dan R menjadi faktor penentu dalam pemilihan metode ini. Dalam penelitian ini, penulis berusaha untuk memilih metode yang mudah dipahami dan diimplementasikan oleh berbagai

pihak, termasuk operator pabrik, manajer produksi, dan analis kualitas. Peta Kendali X dan R menggunakan aturan sederhana untuk menentukan batas kontrol atas dan bawah, sehingga memudahkan dalam mengambil keputusan tentang apakah suatu proses membutuhkan perhatian lebih lanjut atau tidak.

Berikut adalah langkah-langkah pembuatan Peta Kendali X dan R

1. Tentukan ukuran grup ($n = 3, 4, 5, \dots$).
2. Tentukan banyaknya subgrup (k) sedikitnya 20 subgrup.
3. Hitung nilai rata-rata dari setiap subgrup, yaitu \bar{X} .
4. Hitung nilai rata-rata seluruh \bar{X} , yaitu $\bar{\bar{X}}$, yang merupakan center line dari peta kendali \bar{X} .
5. Hitung nilai selisih data terbesar dengan data terkecil dari setiap subgrup, yaitu Range (R).
6. Hitung nilai rata-rata dari seluruh R , yaitu \bar{R} yang merupakan center line dari peta kendali R .
7. Hitung batas kendali dari Peta Kendali \bar{X} :

$$UCL = \bar{\bar{X}} + (A_2 \times \bar{R})$$

$$LCL = \bar{\bar{X}} - (A_2 \times \bar{R})$$

Dimana $A_2 = \frac{3}{d_2\sqrt{n}}$

8. Hitung batas kendali untuk Peta Kendali R

$$UCL = D_4 \times \bar{R}$$

$$LCL = D_3 \times \bar{R}$$

9. Plot data \bar{X} dan R pada peta kendali \bar{X} dan R serta amati apakah data tersebut berada dalam pengendalian atau tidak.

10. Hitung Indeks Kapabilitas Proses (C_p)

Kapabilitas proses digunakan untuk melihat kapabilitas atau kemampuan proses. Indeks kapabilitas proses hanya layak dihitung apabila proses berada dalam pengendalian

$$C_p = \frac{USL - LSL}{6S}$$

Dimana $S = \sqrt{\frac{(N \sum X_i^2) - (\sum X_i)^2}{N(N-1)}}$ atau $S = \frac{R}{d_2}$

Kriteria penilaian :

- Jika $C_p > 1,33$, maka kapabilitas proses sangat baik
- Jika $1,00 \leq C_p \leq 1,33$, maka kapabilitas proses baik
- Jika $C_p < 1,00$, maka kapabilitas proses rendah

11. Hitung Indeks C_{pk} :

$$C_{pk} = \text{Minimum} \{CPU ; CPL\}$$

Dimana : $CPU = \frac{USL - \bar{X}}{3S}$ dan $CPL = \frac{\bar{X} - LSL}{3S}$

Kriteria penilaian :

- Jika $C_{pk} = C_p$, maka proses terjadi ditengah
- Jika $C_{pk} = 1$, maka proses menghasilkan produk yang sesuai dengan spesifikasi
- Jika $C_{pk} < 1$, maka proses menghasilkan produk yang tidak sesuai dengan spesifikasi
- Kondisi Ideal : $C_p > 1,33$ dan $C_p = C_{pk}$

Keterangan : A2 , D3 dan D4 adalah nilai konstanta (Tabel Lampiran)

Bagian 4. Pengolahan data dan analisis hasil

Berikut adalah analisis yang dilakukan menggunakan **Peta Kendali X dan R** :

1. Ukuran group pada penelitian ini adalah $n = 5$, yaitu Tahun 2019,2020,2021,2022,2023.
2. Banyaknya subgroup pada penelitian ini adalah $k = 547$ dengan subgroup pertama adalah ACEH dan subgroup ke 547 adalah Kota Jayapura.
3. Rata-rata nilai setiap subgroup ditunjukkan dengan kolom xbar dan rata-rata nilai keseluruhan adalah sebesar 470799.65
4. Rata-rata range setiap subgroup ditunjukkan dengan kolom R dan rata-rata range keseluruhan adalah sebesar 118731.89

```
121] clx = sum(xbar)/len(df)
      clr = sum(r)/len(df)

      print(clx)
      print(clr)

➞ 470799.6536496352
   118731.89781021897
```

5. Hitung batas atas dan batas bawah Peta Kendali X seperti pada rumus yang dijelaskan di bagian 3 penelitian ini. Berikut adalah tabel batas nya

	Wilayah	Xbar		UCL	LCL	CL
0	ACEH	551386.2	539307.958686	402291.348613	470799.65365	
1	Simeulue	468077.0	539307.958686	402291.348613	470799.65365	
2	Aceh Singkil	499818.2	539307.958686	402291.348613	470799.65365	
3	Aceh Selatan	426874.2	539307.958686	402291.348613	470799.65365	
4	Aceh Tenggara	411271.8	539307.958686	402291.348613	470799.65365	
...
543	Puncak	721617.2	539307.958686	402291.348613	470799.65365	
544	Dogiyai	561557.6	539307.958686	402291.348613	470799.65365	
545	Intan Jaya	727558.0	539307.958686	402291.348613	470799.65365	
546	Deiyai	634704.8	539307.958686	402291.348613	470799.65365	
547	Kota Jayapura	1062390.0	539307.958686	402291.348613	470799.65365	

548 rows × 5 columns

Diperoleh

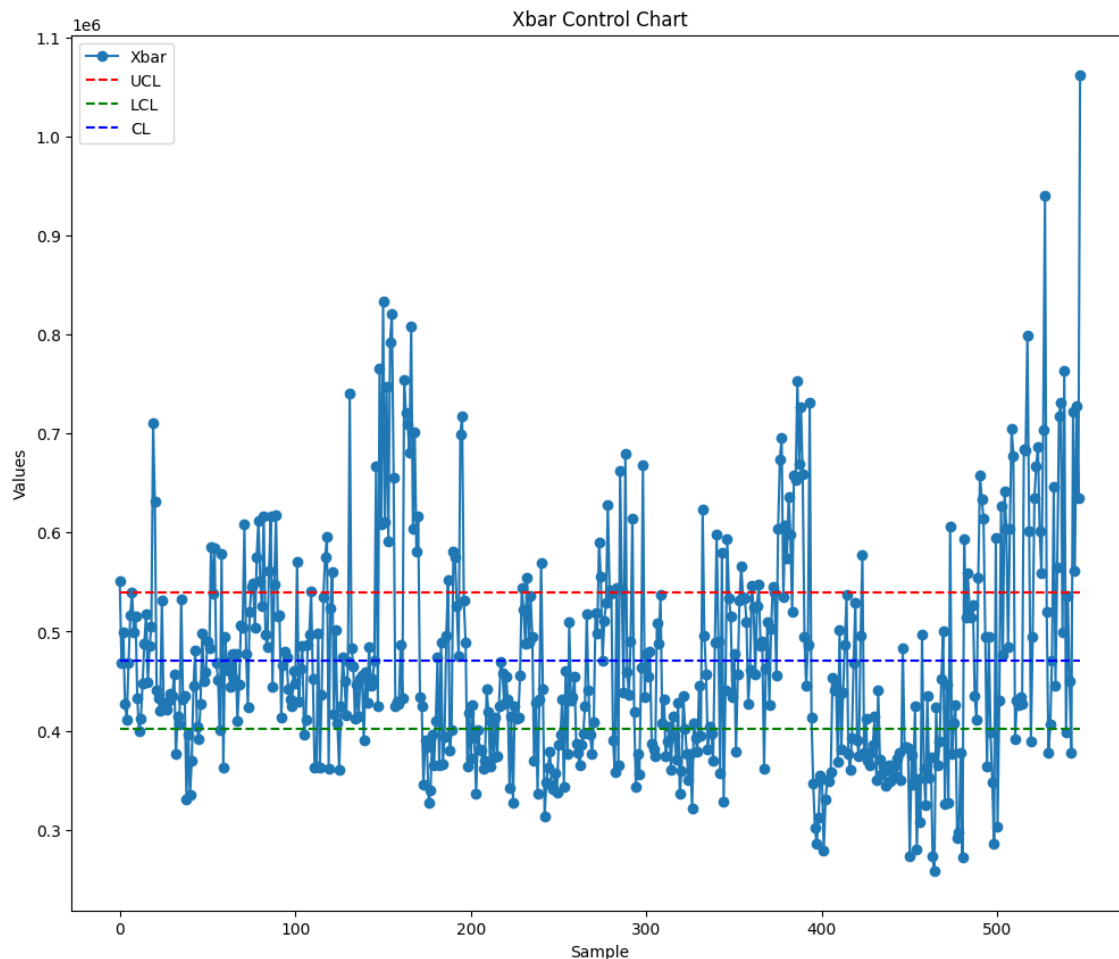
:

UCL : 539307.958686

LCL : 402291.348613

CL : 470799.65365

6. Plot Peta Kendali X



Pada grafik dapat dilihat bahwa terdapat titik dalam sampel yang berada di luar control limit sehingga proses tidak terkendali secara statistik. Setelah dilakukan pengecekan menggunakan python, terdapat 284 sampel yang berada di luar kontrol limit.

```
# Memfilter data yang melebihi batas UCL dan LCL
out_of_control = dfx[(dfx['Xbar'] > dfx['UCL']) | (dfx['Xbar'] < dfx['LCL'])]

# Mencetak data yang melebihi batas UCL dan LCL
print("Data yang melebihi batas UCL dan LCL:")
print(out_of_control)
```

```
Data yang melebihi batas UCL dan LCL:
  Wilayah  Xbar  UCL  LCL  CL
0  ACEH  551386.2  539307.958686  402291.348613  470799.65365
7  Aceh Barat  539352.6  539307.958686  402291.348613  470799.65365
11 Aceh Utara  399327.6  539307.958686  402291.348613  470799.65365
19 Kota Banda Aceh  710726.6  539307.958686  402291.348613  470799.65365
20 Kota Sabang  631774.6  539307.958686  402291.348613  470799.65365
...
543 Puncak  721617.2  539307.958686  402291.348613  470799.65365
544 Dogiyai  561557.6  539307.958686  402291.348613  470799.65365
545 Intan Jaya  727558.0  539307.958686  402291.348613  470799.65365
546 Deiyai  634704.8  539307.958686  402291.348613  470799.65365
547 Kota Jayapura  1062390.0  539307.958686  402291.348613  470799.65365
```

[285 rows x 5 columns]

Sampel ini akan dikeluarkan dari perhitungan kontrol limit. Data yang baru menghasilkan 262 subgroup. Lalu, akan di cek CL, UCL, dan LCL seperti langkah sebelumnya. Diperoleh :

UCL : 532839.921262

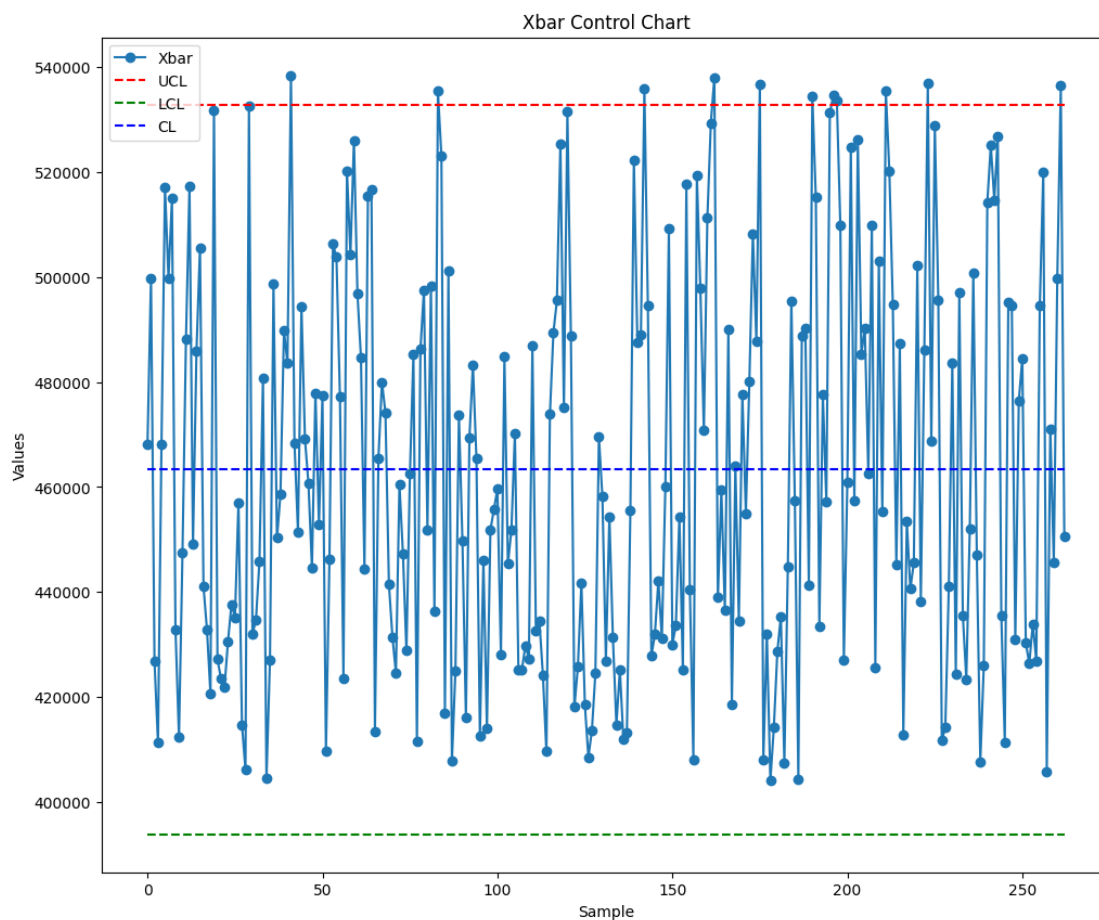
LCL : 393837.786722

CL : 463338.853992

	Wilayah	Xbar	UCL	LCL	CL
0	Simeulue	468077.0	532839.921262	393837.786722	463338.853992
1	Aceh Singkil	499818.2	532839.921262	393837.786722	463338.853992
2	Aceh Selatan	426874.2	532839.921262	393837.786722	463338.853992
3	Aceh Tenggara	411271.8	532839.921262	393837.786722	463338.853992
4	Aceh Timur	468215.2	532839.921262	393837.786722	463338.853992
...
258	Yahukimo	471095.2	532839.921262	393837.786722	463338.853992
259	Tolikara	445640.8	532839.921262	393837.786722	463338.853992
260	Supiori	499795.2	532839.921262	393837.786722	463338.853992
261	Lanny Jaya	536501.0	532839.921262	393837.786722	463338.853992
262	Mamberamo Tengah	450524.4	532839.921262	393837.786722	463338.853992

263 rows × 5 columns

Berikut adalah plot setelah data yang diluar kontrol, dikeluarkan



Terlihat pada grafik bahwa masih terdapat titik dalam sampel yang berada di luar control limit sehingga proses tidak terkendali secara statistik. Setelah dilakukan pengecekan menggunakan python, terdapat 11 sampel yang berada di luar kontrol limit.

Data yang melebihi batas UCL dan LCL:

	Wilayah	Xbar	UCL	LCL	CL
41	Kota Tebing Tinggi	538466.8	532839.921262	393837.786722	463338.853992
83	Musi Rawas Utara	535465.8	532839.921262	393837.786722	463338.853992
142	Kota Tegal	535986.8	532839.921262	393837.786722	463338.853992
162	BANTEN	538053.8	532839.921262	393837.786722	463338.853992
175	Kota Mataram	536844.4	532839.921262	393837.786722	463338.853992
190	Kota Singkawang	534400.6	532839.921262	393837.786722	463338.853992
196	Lamandau	534643.0	532839.921262	393837.786722	463338.853992
197	Seruyan	533672.4	532839.921262	393837.786722	463338.853992
211	Paser	535443.2	532839.921262	393837.786722	463338.853992
223	Poso	536883.0	532839.921262	393837.786722	463338.853992
261	Lanny Jaya	536501.0	532839.921262	393837.786722	463338.853992

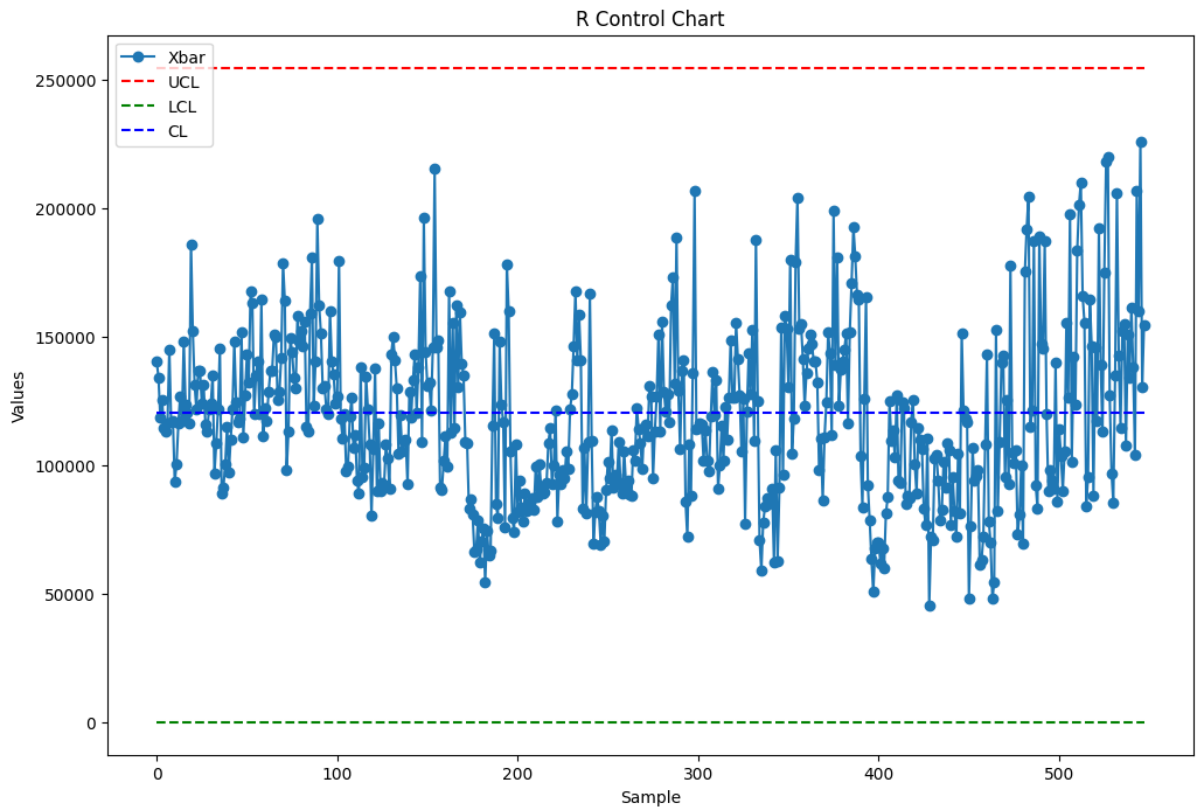
Kesebelas sampel ini akan dikeluarkan dari data, dan akan dilakukan penghitungan Peta Kendali X seperti langkah sebelumnya sampai mendapat grafik yang tidak memiliki nilai diluar batas kendali. Sehingga, dapat disimpulkan bahwa secara keseluruhan, rata-rata garis kemiskinan kabupaten/koya berada diluar batas kendali.

7. Hitung batas atas dan batas bawah Peta Kendali R seperti pada rumus yang dijelaskan di bagian 3 penelitian ini. Berikut adalah tabel batas nya

	Wilayah	R	UCL	LCL	CL
0	ACEH	140599	254636.492563	0.0	120452.456274
1	Simeulue	133954	254636.492563	0.0	120452.456274
2	Aceh Singkil	118474	254636.492563	0.0	120452.456274
3	Aceh Selatan	125458	254636.492563	0.0	120452.456274
4	Aceh Tenggara	114286	254636.492563	0.0	120452.456274
...
543	Puncak	206754	254636.492563	0.0	120452.456274
544	Dogiyai	160102	254636.492563	0.0	120452.456274
545	Intan Jaya	225970	254636.492563	0.0	120452.456274
546	Deiyai	130486	254636.492563	0.0	120452.456274
547	Kota Jayapura	154655	254636.492563	0.0	120452.456274

548 rows × 5 columns

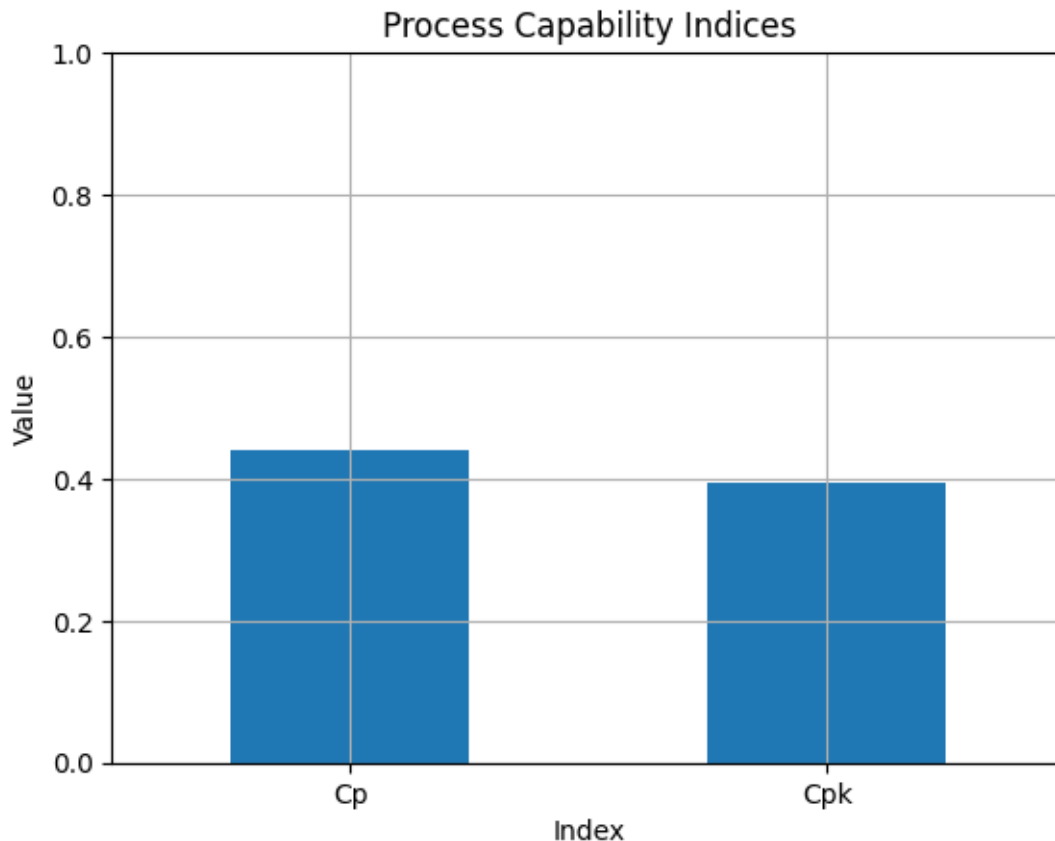
8. Plot Peta Kendali R



Pada grafik dapat dilihat bahwa semua titik sampel berada di dalam control limit sehingga proses terkendali secara statistik. Dengan demikian dapat disimpulkan bahwa variasi subgroup berdasarkan rentangan grup berada di dalam batas kendali

8. Dari python diperoleh nilai Kapabilitas Proses (Cp dan Cpk) sebagai berikut :

Value	
Index	
Cp	0.440977
Cpk	0.392953



Nilai Cp sebesar 0.440977 ternyata kurang dari 1, hal ini menunjukkan kapabilitas proses untuk memenuhi spesifikasi yang ditentukan tergolong rendah.

Nilai Cpk sebesar 0.392953, menunjukkan bahwa proses cenderung mendekati batas spesifikasi bawah atau proses menghasilkan produk yang tidak sesuai dengan spesifikasi

Bagian 5. Penutup

Berdasarkan pembahasan yang telah diuraikan diatas, maka dapat disimpulkan :

1. Hasil analisis statistical quality control (SQC) dengan peta kendali X dan R menunjukkan jumlah sampel yang diluar batas kendali untuk garis kemiskinan sebanyak 273 sampel dari 547 sampel kabupaten/kota (49,9%) tidak memenuhi standar mutu.
2. Nilai proses kapabilitas (Cp) untuk garis kemiskinan sebesar 0.440977. Sedangkan untuk nilai kinerja prosesnya (Cpk) untuk garis kemiskinan sebesar 0.392953. Setelah dilakukan pengolahan data maka diketahui bahwa kapabilitas proses rendah, sehingga membebani kinerja proses yang saat ini diterapkan oleh pemerintah Indonesia.
3. Dari visualisasi dapat dilihat bahwa pada 5 tahun terakhir, Kota Jayapura sebagai kota dengan garis kemiskinan tertinggi secara berturut”. Hal ini menunjukkan perlu adanya bantuan ekstra dari pemerintah untuk menyejahterakan kota ini. Sebaliknya, kota Buton sebagai kota dengan garis kemiskinan terendah selama 4 Tahun terakhir. Hal ini menunjukkan, kota Buton sebagai kota paling sejahtera di Indonesia.

4. Menampilkan tren untuk setiap wilayah melalui line plot akan sulit dilakukan dalam satu plot, karena terdapat 548 wilayah pada data. Sebagai gantinya penulis memilih 5 wilayah secara acak untuk melihat tren garis kemiskinan. Kota/kabupaten yang dipilih adalah ACEH, Bungo, Cilacap, Lombok Barat, Minahasa Utara, Kota Jayapura. Dapat dilihat tren naik garis kemiskinan pada kelima kota/kabupaten ini. Hal ini menunjukkan perlu penanganan serius dari pemerintah agar angka ini tidak terus naik di tahun yang akan datang.

Bagian 6. Lampiran

Link Codes di Google Collab :

https://colab.research.google.com/drive/1SBTT_9ZFfYLv2tOk53gUP91zxH9O0boZ?usp=sharing

Link Data (*Excel) di Drive :

https://drive.google.com/drive/folders/1JpSQQXBHERr7FJoUGhMkA5PsLELJSe4l?usp=drive_link

Tabel Kontrol Limit

■ APPENDIX VI																	
Factors for Constructing Variables Control Charts																	
Observations in Sample, <i>n</i>	Chart for Averages					Chart for Standard Deviations						Chart for Ranges					
	Factors for Control Limits			Factors for Center Line		Factors for Control Limits				Factors for Center Line		Factors for Control Limits					
	<i>A</i>	<i>A</i> ₂	<i>A</i> ₃	<i>c</i> ₄	1/ <i>c</i> ₄	<i>B</i> ₃	<i>B</i> ₄	<i>B</i> ₅	<i>B</i> ₆	<i>d</i> ₂	1/ <i>d</i> ₂	<i>d</i> ₃	<i>D</i> ₁	<i>D</i> ₂	<i>D</i> ₃	<i>D</i> ₄	
2	2.121	1.880	2.659	0.7979	1.2533	0	3.267	0	2.606	1.128	0.8865	0.853	0	3.686	0	3.267	
3	1.732	1.023	1.954	0.8862	1.1284	0	2.568	0	2.276	1.693	0.5907	0.888	0	4.358	0	2.574	
4	1.500	0.729	1.628	0.9213	1.0854	0	2.266	0	2.088	2.059	0.4857	0.880	0	4.698	0	2.282	
5	1.342	0.577	1.427	0.9400	1.0638	0	2.089	0	1.964	2.326	0.4299	0.864	0	4.918	0	2.114	
6	1.225	0.483	1.287	0.9515	1.0510	0.030	1.970	0.029	1.874	2.534	0.3946	0.848	0	5.078	0	2.004	
7	1.134	0.419	1.182	0.9594	1.0423	0.118	1.882	0.113	1.806	2.704	0.3698	0.833	0.204	5.204	0.076	1.924	
8	1.061	0.373	1.099	0.9650	1.0363	0.185	1.815	0.179	1.751	2.847	0.3512	0.820	0.388	5.306	0.136	1.864	
9	1.000	0.337	1.032	0.9693	1.0317	0.239	1.761	0.232	1.707	2.970	0.3367	0.808	0.547	5.393	0.184	1.816	
10	0.949	0.308	0.975	0.9727	1.0281	0.284	1.716	0.276	1.669	3.078	0.3249	0.797	0.687	5.469	0.223	1.777	
11	0.905	0.285	0.927	0.9754	1.0252	0.321	1.679	0.313	1.637	3.173	0.3152	0.787	0.811	5.535	0.256	1.744	
12	0.866	0.266	0.886	0.9776	1.0229	0.354	1.646	0.346	1.610	3.258	0.3069	0.778	0.922	5.594	0.283	1.717	
13	0.832	0.249	0.850	0.9794	1.0210	0.382	1.618	0.374	1.585	3.336	0.2998	0.770	1.025	5.647	0.307	1.693	
14	0.802	0.235	0.817	0.9810	1.0194	0.406	1.594	0.399	1.563	3.407	0.2935	0.763	1.118	5.696	0.328	1.672	
15	0.775	0.223	0.789	0.9823	1.0180	0.428	1.572	0.421	1.544	3.472	0.2880	0.756	1.203	5.741	0.347	1.653	
16	0.750	0.212	0.763	0.9835	1.0168	0.448	1.552	0.440	1.526	3.532	0.2831	0.750	1.282	5.782	0.363	1.637	
17	0.728	0.203	0.739	0.9845	1.0157	0.466	1.534	0.458	1.511	3.588	0.2787	0.744	1.356	5.820	0.378	1.622	
18	0.707	0.194	0.718	0.9854	1.0148	0.482	1.518	0.475	1.496	3.640	0.2747	0.739	1.424	5.856	0.391	1.608	
19	0.688	0.187	0.698	0.9862	1.0140	0.497	1.503	0.490	1.483	3.689	0.2711	0.734	1.487	5.891	0.403	1.597	
20	0.671	0.180	0.680	0.9869	1.0133	0.510	1.490	0.504	1.470	3.735	0.2677	0.729	1.549	5.921	0.415	1.585	
21	0.655	0.173	0.663	0.9876	1.0126	0.523	1.477	0.516	1.459	3.778	0.2647	0.724	1.605	5.951	0.425	1.575	
22	0.640	0.167	0.647	0.9882	1.0119	0.534	1.466	0.528	1.448	3.819	0.2618	0.720	1.659	5.979	0.434	1.566	
23	0.626	0.162	0.633	0.9887	1.0114	0.545	1.455	0.539	1.438	3.858	0.2592	0.716	1.710	6.006	0.443	1.557	
24	0.612	0.157	0.619	0.9892	1.0109	0.555	1.445	0.549	1.429	3.895	0.2567	0.712	1.759	6.031	0.451	1.548	
25	0.600	0.153	0.606	0.9896	1.0105	0.565	1.435	0.559	1.420	3.931	0.2544	0.708	1.806	6.056	0.459	1.541	

Codes :

```
import numpy as np # linear algebra
import os
import pandas as pd
import matplotlib.pyplot as plt
```



```

from google.colab import drive
drive.mount('/content/drive')

# Ganti 'namafile.xlsx' dengan nama file Anda di Google Drive
data = '/content/drive/MyDrive/Stat QC/Garis Kemiskinan Menurut
Kabupaten_Kota 2019-2023.xlsx'
ukuran = '/content/drive/MyDrive/Stat QC/x-bar-r.xlsx'
# Membaca file Excel
df = pd.read_excel(data)
df1 = pd.read_excel(ukuran)

dfs = pd.read_excel(data, usecols=['Tahun 2019', 'Tahun 2020', 'Tahun
2021', 'Tahun 2022', 'Tahun 2023'])

# Menampilkan beberapa baris pertama dari dataframe
print(df.head())

# Menampilkan beberapa baris pertama dari tabel ukuran yang akan
digunakan
print(df1.head())

df.describe()

# Memilih beberapa wilayah random untuk ditampilkan
selected_wilayah = df.iloc[[0, 100, 200, 300, 400, 547]]
# Plot line plot
plt.figure(figsize=(14, 8))
for i, row in selected_wilayah.iterrows():
    plt.plot(['Tahun 2019', 'Tahun 2020', 'Tahun 2021', 'Tahun 2022',
'Tahun 2023'], row[1:], marker='o', label=row['Wilayah'])

plt.title('Tren Garis Kemiskinan dari 2019 hingga 2023 untuk Beberapa
Wilayah')
plt.xlabel('Tahun')
plt.ylabel('Garis Kemiskinan (Rupiah/kapita/bulan)')
plt.legend(title='Wilayah')
plt.show()

# Mengubah data untuk box plot
df_melted = pd.melt(df, id_vars=['Wilayah'], value_vars=['Tahun 2019',
'Tahun 2020', 'Tahun 2021', 'Tahun 2022', 'Tahun 2023'],
var_name='Tahun', value_name='Garis Kemiskinan')

plt.figure(figsize=(14, 8))
sns.boxplot(x='Tahun', y='Garis Kemiskinan', data=df_melted)

```

```

plt.title('Distribusi Garis Kemiskinan dari 2019 hingga 2023')
plt.xlabel('Tahun')
plt.ylabel('Garis Kemiskinan (Rupiah/kapita/bulan)')
plt.show()

# Mengidentifikasi wilayah dengan angka garis kemiskinan terendah dan tertinggi tiap tahunnya
years = ['Tahun 2019', 'Tahun 2020', 'Tahun 2021', 'Tahun 2022', 'Tahun 2023']
lowest_highest = {'Year': [], 'Lowest': [], 'Highest': [], 'Lowest Value': [], 'Highest Value': []}

for year in years:
    lowest = df.loc[df[year].idxmin()]
    highest = df.loc[df[year].idxmax()]
    lowest_highest['Year'].append(year)
    lowest_highest['Lowest'].append(lowest['Wilayah'])
    lowest_highest['Highest'].append(highest['Wilayah'])
    lowest_highest['Lowest Value'].append(lowest[year])
    lowest_highest['Highest Value'].append(highest[year])

# Membuat DataFrame dari hasil identifikasi
df_low_high = pd.DataFrame(lowest_highest)

# Menampilkan wilayah dengan garis kemiskinan terendah dan tertinggi tiap tahun
print(df_low_high)

# Visualisasi
plt.figure(figsize=(14, 8))

# Bar plot untuk wilayah dengan angka garis kemiskinan terendah
plt.subplot(1, 2, 1)
plt.bar(df_low_high['Year'], df_low_high['Lowest Value'],
color='green')
plt.xticks(rotation=45)
plt.title('Wilayah dengan Garis Kemiskinan Terendah Tiap Tahun')
plt.xlabel('Tahun')
plt.ylabel('Garis Kemiskinan (Rupiah/kapita/bulan)')
for i, v in enumerate(df_low_high['Lowest Value']):
    plt.text(i, v + 1000, df_low_high['Lowest'][i], ha='center',
color='black')

# Bar plot untuk wilayah dengan angka garis kemiskinan tertinggi
plt.subplot(1, 2, 2)
plt.bar(df_low_high['Year'], df_low_high['Highest Value'], color='red')
plt.xticks(rotation=45)

```

```

plt.title('Wilayah dengan Garis Kemiskinan Tertinggi Tiap Tahun')
plt.xlabel('Tahun')
plt.ylabel('Garis Kemiskinan (Rupiah/kapita/bulan)')
for i, v in enumerate(df_low_high['Highest Value']):
    plt.text(i, v + 1000, df_low_high['Highest'][i], ha='center',
color='black')

plt.tight_layout()
plt.show()

xbar = []
r = []
for i in range(len(df)):
    x = (df['Tahun 2019'][i]+df['Tahun 2020'][i]+df['Tahun
2021'][i]+df['Tahun 2022'][i]+df['Tahun 2023'][i])/5
    rang = max(dfs.loc[i]) - min(dfs.loc[i])
    xbar.append(x)
    r.append(rang)

data_xr = {'Wilayah':df['Wilayah'],'Tahun 2019':df['Tahun 2019'],'Tahun
2020':df['Tahun 2020'],'Tahun 2021':df['Tahun 2021'],'Tahun
2022':df['Tahun 2022'],'Tahun 2023':df['Tahun
2023'],'Xbar':xbar,'R':r,}
dxr = pd.DataFrame(data_xr)
dxr

clx = sum(xbar)/len(df)
clr = sum(r)/len(df)

print(clx)
print(clr)

uclx,lclx = [],[]
Uclx = clx + df1.loc[3]['A2']*clr
Lclx = clx - df1.loc[3]['A2']*clr

for k in range(len(df)):
    uclx.append(Uclx)
    lclx.append(Lclx)

data_xbar =
{'Wilayah':df['Wilayah'],'Xbar':xbar,'UCL':uclx,'LCL':lclx,'CL':clx}
dfx = pd.DataFrame(data_xbar)
dfx

# Membuat plot

```

```

plt.figure(figsize=(12, 10))

# Plot Xbar
plt.plot(dfx['Xbar'], marker='o', label='Xbar')

# Plot UCL
plt.plot(dfx['UCL'], linestyle='--', color='r', label='UCL')

# Plot LCL
plt.plot(dfx['LCL'], linestyle='--', color='g', label='LCL')

# Plot CL
plt.plot(dfx['CL'], linestyle='--', color='b', label='CL')

# Menambahkan judul dan label
plt.title('Xbar Control Chart')
plt.xlabel('Sample')
plt.ylabel('Values')

# Menampilkan legenda
plt.legend()

# Menampilkan plot
chart_x=plt.show()

# Memfilter data yang melebihi batas UCL dan LCL
out_of_control = dfx[(dfx['Xbar'] > dfx['UCL']) | (dfx['Xbar'] <
dfx['LCL'])]

# Mencetak data yang melebihi batas UCL dan LCL
print("Data yang melebihi batas UCL dan LCL:")
print(out_of_control)

# Menampilkan isi kolom 'Wilayah' yang out of control
print("Wilayah yang out of control:")
print(out_of_control['Wilayah'].values)

dfxc = dxr
# Menghapus baris data yang di luar kendali
dfx_cleaned = dfxc.drop(out_of_control.index)

# Mengatur ulang indeks
dfx_cleaned.reset_index(drop=True, inplace=True)

# Dataframe setelah penghapusan
print("\nData setelah penghapusan:")
print(dfx_cleaned)

```

```

clx = sum(dfx_cleaned['Xbar']) / len(dfx_cleaned)
clr = sum(dfx_cleaned['R']) / len(dfx_cleaned)

print(clx)
print(clr)

uclx,lclx = [],[]
Uclx = clx + df1.loc[3]['A2']*clr
Lclx = clx - df1.loc[3]['A2']*clr

for k in range(len(dfx_cleaned)):
    uclx.append(Uclx)
    lclx.append(Lclx)

data_xbar =
{'Wilayah':dfx_cleaned['Wilayah'],'Xbar':dfx_cleaned['Xbar'],'UCL':uclx
,'LCL':lclx,'CL':clx}
dfxc = pd.DataFrame(data_xbar)
dfxc

# Membuat plot
plt.figure(figsize=(12, 10))

# Plot Xbar
plt.plot(dfxc['Xbar'], marker='o', label='Xbar')

# Plot UCL
plt.plot(dfxc['UCL'], linestyle='--', color='r', label='UCL')

# Plot LCL
plt.plot(dfxc['LCL'], linestyle='--', color='g', label='LCL')

# Plot CL
plt.plot(dfxc['CL'], linestyle='--', color='b', label='CL')

# Menambahkan judul dan label
plt.title('Xbar Control Chart')
plt.xlabel('Sample')
plt.ylabel('Values')

# Menampilkan legenda
plt.legend()

# Menampilkan plot
chart_x=plt.show()

```

```

# Memfilter data yang melebihi batas UCL dan LCL
out_of_control = dfxc[(dfxc['Xbar'] > dfxc['UCL']) | (dfxc['Xbar'] <
dfxc['LCL'])]

# Mencetak data yang melebihi batas UCL dan LCL
print("Data yang melebihi batas UCL dan LCL:")
print(out_of_control)

# Menampilkan isi kolom 'Wilayah' yang out of control
print("Wilayah yang out of control:")
print(out_of_control['Wilayah'].values)

uclr,lclr = [],[]
Uclr = clr * df1.loc[3]['D4']
Lclr = clr * df1.loc[3]['D3']

for l in range(len(df)):
    uclr.append(Uclr)
    lclr.append(Lclr)

data_r = {'Wilayah':df['Wilayah'],'R':r,'UCL':uclr,'LCL':lclr,'CL':clr}
dfr = pd.DataFrame(data_r)
dfr

# Membuat plot
plt.figure(figsize=(12, 8))

# Plot Xbar
plt.plot(dfr['R'], marker='o', label='Xbar')

# Plot UCL
plt.plot( dfr['UCL'], linestyle='--', color='r', label='UCL')

# Plot LCL
plt.plot( dfr['LCL'], linestyle='--', color='g', label='LCL')

# Plot CL
plt.plot( dfr['CL'], linestyle='--', color='b', label='CL')

# Menambahkan judul dan label
plt.title('R Control Chart')
plt.xlabel('Sample')
plt.ylabel('Values')

# Menampilkan legenda
plt.legend()

```

```

# Menampilkan plot
chart_r=plt.show()

s = clr/df1.loc[3]['d2']
usl = 539307.958686
lsl = 402291.348613

print('Mean : ',clx)
print('Sigma : ',s)
print('Upper Specification Limit : ',usl)
print('Lower Specification Limit : ',lsl)

cp = (usl - lsl) / (6 * s)
cpu = (usl - clx) / (3 * s)
cpl = (clx - lsl) / (3 * s)

print('Capability Process : ',cp)
print('Capability Process (Upper) : ',cpu)
print('Capability Process (Lower) : ',cpl)

cpk = min(cpu,cpl)
cr = 1/cp

print('Capability Process Index : ',cpk)
print(f'Capability Ratio : {cr*100} %')

# Compile the capability indices into a DataFrame
capability_indices = pd.DataFrame({
    'Index': ['Cp', 'Cpk'],
    'Value': [cp, cpk]
})

# Display the process capability indices
capability_indices.set_index('Index', inplace=True)
capability_indices.plot(kind='bar', legend=False)
plt.title('Process Capability Indices')
plt.ylabel('Value')
plt.xlabel('Index')
plt.xticks(rotation=0)
plt.ylim(0, 1)
plt.grid(True)
plt.show()

capability_indices

```