

tugas-eda-2

November 1, 2024

1 Tren Pengeluaran Prioritas Anggaran Provinsi DKI Jakarta

Kelompok 10

1. Muhammad Igin Adigholib (2301125)
2. Rahmat Taufik Al-Hidayah (2300414)
3. Anugrah Bayu Satrio (2300624)
4. Muhamad Khafabillah Sopian (2307824)
5. Remy Putra Nur Laksana (2309578)

2 Pendahuluan

Eksplorasi Data Analysis (EDA) merupakan langkah awal yang penting dalam memahami karakteristik suatu dataset. EDA berfungsi sebagai jembatan antara pengumpulan data dan analisis yang lebih dalam, memungkinkan para peneliti untuk mengeksplorasi dan memahami data mereka dengan cara yang lebih komprehensif. Dengan menggunakan teknik visualisasi dan statistik deskriptif, EDA dapat mengungkapkan pola, tren, dan anomali dalam data, yang semuanya krusial untuk pengambilan keputusan berbasis data 1 2 .

Dalam konteks analisis anggaran DKI Jakarta, EDA bertujuan untuk mengidentifikasi tren pengeluaran dan prioritas alokasi dana untuk berbagai kegiatan. Analisis ini sangat relevan, terutama dalam konteks pengelolaan anggaran daerah yang harus memenuhi beragam kebutuhan masyarakat. Dengan memanfaatkan metode EDA, kita dapat menggali informasi terkait bagaimana anggaran dialokasikan ke berbagai sektor, seperti pendidikan, kesehatan, dan infrastruktur, serta bagaimana pengeluaran bervariasi dari tahun ke tahun 2 .

Dengan pemahaman yang lebih mendalam tentang distribusi dan penggunaan anggaran, diharapkan analisis ini dapat memberikan gambaran yang jelas tentang bagaimana pemerintah daerah mengarahkan sumber daya untuk pembangunan dan pelayanan publik. Hal ini penting untuk memastikan bahwa pengelolaan anggaran tidak hanya efisien tetapi juga efektif dalam memenuhi kebutuhan masyarakat 2 .

Referensi

- [1] A. S. Rao, B. V. Vardhan and H. Shaik, “Role of Exploratory Data Analysis in Data Science,” 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 1457-1461, doi: 10.1109/ICCES51350.2021.9488986. Diakses pada 1 November 2024.
- [2] P. Iliopoulos dan K. De Witte, “The expenditure composition and trade-offs in local government budgets,” Socio-Economic Planning Sciences, vol. 93, 2024, art. 101900, doi:

10.1016/j.seps.2024.101900. Diakses pada 1 November 2024.

3 Sumber Data

https://katalog.data.go.id/dataset/data-apbd-provinsi-dki-jakarta?kategori=&q=apbd&sort=prioritas_tahun%20desc
https://katalog.data.go.id/dataset/data-anggaran-pendapatan-belanja-dki-jakarta-tahun-2014?kategori=&q=apbd&sort=prioritas_tahun%20desc
https://katalog.data.go.id/dataset/data-perencanaan-apbd-2018?kategori=&q=APBD%20&sort=prioritas_tahun%20desc
https://katalog.data.go.id/dataset/data-capaian-pelaksanaan-kegiatan-apbd-2017?kategori=&q=APBD%20&sort=prioritas_tahun%20desc
https://katalog.data.go.id/dataset/data-apbd-provinsi-dki-jakarta-tahun-2020?currentPage=0&facet.field=organization,kategori,prioritas_tahun,tags,res_format&facet.limit=500&rows=50
https://katalog.data.go.id/dataset/data-apbd-provinsi-dki-jakarta-tahun-2019?kategori=&q=APBD%20&sort=prioritas_tahun%20desc https://katalog.data.go.id/dataset/data-anggaran-belanja-dki-jakarta-per-kegiatan?currentPage=0&facet.field=organization,kategori,prioritas_tahun,tags,res_format&facet.limit=500&rows=50
https://katalog.data.go.id/dataset/data-apbd-provinsi-dki-jakarta-tahun-2016?currentPage=0&facet.field=organization,kategori,prioritas_tahun,tags,res_format&facet.limit=500&rows=50
https://katalog.data.go.id/dataset/data-apbd-penetapan-2015-dki-jakarta?kategori=&q=apbd&sort=prioritas_tahun%20desc

4 Perpustakaan

Mengimpor library yang diperlukan untuk manipulasi data (pandas, numpy), visualisasi (matplotlib, seaborn).

```
[ ]: # Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.ticker import FuncFormatter
from os.path import join
from IPython.core.display import display, HTML
```

5 Impor Drive

Menghubungkan Google Colab dengan Google Drive untuk mengakses file yang tersimpan di dalamnya.

```
[ ]: from google.colab import drive
drive.mount('/content/drive')
dir = "/content/drive/MyDrive/Data Mining(EDA)/"
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

6 Akses Drive

Memuat data anggaran belanja dan APBD DKI Jakarta dari beberapa file CSV ke dalam DataFrame, termasuk data realisasi belanja 2014 hingga APBD 2020, serta capaian pelaksanaan kegiatan APBD 2017.

```
[ ]: # Memuat data dari CSV
belanja = pd.read_csv(join(dir, "Filedata Data Anggaran Belanja DKI Jakarta Per_
↳Kegiatan.csv"))
belanja2014 = pd.read_csv(join(dir, "Filedata Data Realisasi Belanja 2014_
↳Pemerintah Provinsi DKI Jakarta.csv"))
apbd = pd.read_csv(join(dir, "Data APBD Provinsi DKI Jakarta.csv"))
apbd2015 = pd.read_csv(join(dir, "Data APBD Penetapan 2015 DKI Jakarta (1).
↳csv"))
apbd2016 = pd.read_csv(join(dir, " Data APBD Provinsi DKI Jakarta Tahun 2016.
↳csv"))
apbd2017 = pd.read_csv(join(dir, "Data APBD Provinsi DKI Jakarta Tahun 2017.
↳csv"))
apbd2018 = pd.read_csv(join(dir, "Data Perencanaan APBD 2018.csv"))
apbd2019 = pd.read_csv(join(dir, "Data APBD Provinsi DKI Jakarta Tahun 2019.
↳csv"))
apbd2020 = pd.read_csv(join(dir, "Data APBD Provinsi DKI Jakarta Tahun 2020 (1).
↳csv"))
pelaksanaan = pd.read_csv(join(dir, "Data Capaian Pelaksanaan Kegiatan APBD_
↳2017 (2).csv"))
```

```
<ipython-input-93-c692b1202934>:9: DtypeWarning: Columns (6,8) have mixed types.
Specify dtype option on import or set low_memory=False.
```

```
apbd2019 = pd.read_csv(join(dir, "Data APBD Provinsi DKI Jakarta Tahun
2019.csv"))
```

```
<ipython-input-93-c692b1202934>:10: DtypeWarning: Columns (6,8) have mixed
types. Specify dtype option on import or set low_memory=False.
```

```
apbd2020 = pd.read_csv(join(dir, "Data APBD Provinsi DKI Jakarta Tahun 2020
(1).csv"))
```

7 Penyesuaian Nama Atribut dan Nama Field Pendidikan

Mengganti nama kolom untuk konsistensi tambahan pada beberapa dataset

```
[ ]: # Mengubah nama kolom menjadi konsisten di seluruh dataset
apbd2015.rename(columns={'urusannama': 'nama_urusan_program', 'anggaran':
↳'nilai_anggaran'}, inplace=True)
pelaksanaan.rename(columns={'urusan_nama': 'nama_urusan_program', 'anggaran':
↳'nilai_anggaran'}, inplace=True)
belanja2014.rename(columns={'nama_urusan': 'nama_urusan_program', 'anggaran':
↳'nilai_anggaran'}, inplace=True)
```

```

apbd2018.rename(columns={'nama_urusan': 'nama_urusan_program', 'anggaran':
↳ 'nilai_anggaran'}, inplace=True)

# Mengubah "Pendidikan" menjadi uppercase pada masing-masing dataset
apbd2015['nama_urusan_program'] = apbd2015['nama_urusan_program'].apply(lambda
↳ x: x.replace('Pendidikan', 'PENDIDIKAN') if 'Pendidikan' in x else x)
pelaksanaan['nama_urusan_program'] = pelaksanaan['nama_urusan_program'].
↳ apply(lambda x: x.replace('Pendidikan', 'PENDIDIKAN') if 'Pendidikan' in x
↳ else x)
apbd2018['nama_urusan_program'] = apbd2018['nama_urusan_program'].apply(lambda
↳ x: x.replace('Pendidikan', 'PENDIDIKAN') if 'Pendidikan' in x else x)

# Memilih kolom yang diperlukan
data_belanja = belanja2014[['nama_urusan_program', 'nilai_anggaran',
↳ 'periode_data']]
data_apbd2015 = apbd2015[['nama_urusan_program', 'nilai_anggaran',
↳ 'periode_data']]
data_apbd2016 = apbd2016[['nama_urusan_program', 'nilai_anggaran',
↳ 'periode_data']]
data_apbd2017 = apbd2017[['nama_urusan_program', 'nilai_anggaran',
↳ 'periode_data']]
data_apbd2018 = apbd2018[['nama_urusan_program', 'nilai_anggaran',
↳ 'periode_data']]
data_apbd2019 = apbd2019[['nama_urusan_program', 'nilai_anggaran',
↳ 'periode_data']]
data_apbd2020 = apbd2020[['nama_urusan_program', 'nilai_anggaran',
↳ 'periode_data']]
data_apbd = apbd[['nilai_anggaran', 'periode_data']]
data_pelaksanaan = pelaksanaan[['nama_urusan_program', 'nilai_anggaran',
↳ 'periode_data']]

```

Tiga kode ini memfilter data “PENDIDIKAN” dari dataset APBD Penetapan 2015 setelah mengganti semua kemunculan “Pendidikan” menjadi uppercase untuk konsistensi, hanya menampilkan kolom ‘nama_urusan_program’, ‘nilai_anggaran’, dan ‘periode_data’.

```

[ ]: # Memfilter data khusus untuk "PENDIDIKAN" dan hanya menampilkan tiga kolom
data_pendidikan_2015 = apbd2015.loc[apbd2015['nama_urusan_program'] ==
↳ 'PENDIDIKAN',
                                  ['nama_urusan_program', 'nilai_anggaran',
↳ 'periode_data']]

# Menghapus koma dari nilai_anggaran dan mengubah tipe data menjadi integer
data_pendidikan_2015['nilai_anggaran'] = (
    data_pendidikan_2015['nilai_anggaran']
    .replace(',', '', regex=True) # Menghapus koma

```

```

        .astype(int) # Mengonversi ke integer
    )

    # Menampilkan hasilnya
    print("Data Pendidikan APBD Penetapan 2015:")
    print(data_pendidikan_2015.head())

```

Data Pendidikan APBD Penetapan 2015:

	nama_urusan_program	nilai_anggaran	periode_data
0	PENDIDIKAN	1500000000	2015
1	PENDIDIKAN	1000000000	2015
2	PENDIDIKAN	1000000000	2015
3	PENDIDIKAN	1000000000	2015
4	PENDIDIKAN	1000000000	2015

```

[ ]: # Memfilter data khusus untuk "PENDIDIKAN" pada dataset Capaian Pelaksanaan
      ↳ Kegiatan APBD 2017
data_pendidikan_2017 = pelaksanaan.loc[pelaksanaan['nama_urusan_program'] ==
      ↳ 'PENDIDIKAN', ['nama_urusan_program', 'nilai_anggaran', 'periode_data']]
print("\nData Pendidikan Capaian Pelaksanaan Kegiatan APBD 2017:")
print(data_pendidikan_2017.head())

```

Data Pendidikan Capaian Pelaksanaan Kegiatan APBD 2017:

	nama_urusan_program	nilai_anggaran	periode_data
127086	PENDIDIKAN	10820000	2017
127087	PENDIDIKAN	10820000	2017
127088	PENDIDIKAN	10820000	2017
127089	PENDIDIKAN	6890000	2017
127090	PENDIDIKAN	6890000	2017

```

[ ]: # Memfilter data khusus untuk "PENDIDIKAN" pada dataset Perencanaan APBD 2018
data_pendidikan_2018 = apbd2018.loc[apbd2018['nama_urusan_program'] ==
      ↳ 'PENDIDIKAN', ['nama_urusan_program', 'nilai_anggaran', 'periode_data']]
print("\nData Pendidikan Perencanaan APBD 2018:")
print(data_pendidikan_2018.head())

```

Data Pendidikan Perencanaan APBD 2018:

	nama_urusan_program	nilai_anggaran	periode_data
3083	PENDIDIKAN	51350000.0	2018
3084	PENDIDIKAN	154050000.0	2018
3124	PENDIDIKAN	363009000.0	2018
3125	PENDIDIKAN	14151000.0	2018
3126	PENDIDIKAN	11032350.0	2018

8 Penggabungan Semua Dataset

Kode ini menggabungkan berbagai dataset terkait anggaran dan belanja DKI Jakarta serta memfilter untuk menampilkan hanya data sektor pendidikan dalam huruf uppercase, beserta kolom 'nilai_anggaran' dan 'periode_data'.

```
[ ]: # Menggabungkan dataset, kecuali yang disebutkan
data_combined = pd.concat([
    belanja2014[['nama_urusan_program', 'nilai_anggaran', 'periode_data']], #
    ↳Dataset Belanja 2014
    data_pendidikan_2015[['nama_urusan_program', 'nilai_anggaran',
    ↳'periode_data']], # Dataset APBD 2015 dengan pendidikan uppercase
    apbd2016[['nama_urusan_program', 'nilai_anggaran', 'periode_data']], #
    ↳Dataset APBD 2016
    apbd2017[['nama_urusan_program', 'nilai_anggaran', 'periode_data']], #
    ↳Dataset APBD 2017
    apbd2018[['nama_urusan_program', 'nilai_anggaran', 'periode_data']], #
    ↳Dataset APBD 2018 dengan pendidikan uppercase
    apbd2019[['nama_urusan_program', 'nilai_anggaran', 'periode_data']], #
    ↳Dataset APBD 2019
    apbd2020[['nama_urusan_program', 'nilai_anggaran', 'periode_data']], #
    ↳Dataset APBD 2020
    # pelaksanaan[['nama_urusan_program', 'nilai_anggaran', 'periode_data']] #
    ↳Dataset Capaian Pelaksanaan Kegiatan
], ignore_index=True)

# Memfilter data untuk sektor pendidikan
data_pendidikan = data_combined[data_combined['nama_urusan_program'] ==
    ↳'PENDIDIKAN'][['nama_urusan_program', 'nilai_anggaran', 'periode_data']]

# Menampilkan hasil pemfilteran
print("Data Gabungan Sektor Pendidikan:")
print(data_pendidikan.head())
```

Data Gabungan Sektor Pendidikan:

	nama_urusan_program	nilai_anggaran	periode_data
8359	PENDIDIKAN	6077500.0	2014
8360	PENDIDIKAN	6160000.0	2014
8361	PENDIDIKAN	0.0	2014
8362	PENDIDIKAN	2860000.0	2014
8363	PENDIDIKAN	22099220.0	2014

Kami tidak menggunakan dataset pada tahun 2013 (Filed Data Anggaran Belanja DKI Jakarta Per Kegiatan.csv), tahun 2017 (Data Capaian Pelaksanaan Kegiatan APBD 2017 (2).csv), dan 2023 (Data APBD Provinsi DKI Jakarta.csv) disebabkan oleh keterbatasan akses informasi yang tersedia. Data yang tidak lengkap atau tidak konsisten dapat mempengaruhi akurasi dan integritas analisis. Oleh karena itu, untuk menjaga fokus dan keandalan analisis, hanya data dari tahun 2014 hingga 2020 yang

digunakan, di mana informasi lebih komprehensif dan dapat memberikan wawasan yang lebih jelas mengenai tren anggaran pendidikan di DKI Jakarta

9 Proses Untuk Data yang Bernilai Null dan Nol

```
[ ]: # Memfilter data untuk sektor pendidikan
data_pendidikan = data_combined[data_combined['nama_urusan_program'] == 'Pendidikan']

# Menghitung jumlah nilai null di kolom 'nilai_anggaran'
jumlah_null = data_pendidikan['nilai_anggaran'].isnull().sum()

# Menghitung jumlah nilai nol di kolom 'nilai_anggaran'
jumlah_nol = (data_pendidikan['nilai_anggaran'] == 0).sum()

# Menampilkan hasil
print(f"Jumlah nilai null dalam sektor Pendidikan: {jumlah_null}")
print(f"Jumlah nilai nol dalam sektor Pendidikan: {jumlah_nol}")
```

Jumlah nilai null dalam sektor Pendidikan: 2
Jumlah nilai nol dalam sektor Pendidikan: 180

```
[ ]: # Menghapus nilai null
data_pendidikan = data_pendidikan.dropna(subset=['nilai_anggaran'])

# Menghapus nilai nol
data_pendidikan = data_pendidikan[data_pendidikan['nilai_anggaran'] != 0]

# Memeriksa nilai null dan nol setelah penghapusan
null_count = data_pendidikan['nilai_anggaran'].isnull().sum()
zero_count = (data_pendidikan['nilai_anggaran'] == 0).sum()

# Menampilkan jumlah nilai null dan nol
print(f"Jumlah nilai null setelah penghapusan: {null_count}")
print(f"Jumlah nilai nol setelah penghapusan: {zero_count}\n")

# Menampilkan hasil setelah menghapus nilai null dan nol
print("Data Pendidikan setelah menghapus nilai null dan nol:")
print(data_pendidikan)
```

Jumlah nilai null setelah penghapusan: 0
Jumlah nilai nol setelah penghapusan: 0

Data Pendidikan setelah menghapus nilai null dan nol:

	nama_urusan_program	nilai_anggaran	periode_data
8359	PENDIDIKAN	6.077500e+06	2014
8360	PENDIDIKAN	6.160000e+06	2014

8362	PENDIDIKAN	2.860000e+06	2014
8363	PENDIDIKAN	2.209922e+07	2014
8364	PENDIDIKAN	1.100000e+07	2014
...
295182	PENDIDIKAN	4.872250e+09	2020
295183	PENDIDIKAN	1.106800e+10	2020
295289	PENDIDIKAN	1.260000e+12	2020
295301	PENDIDIKAN	1.630000e+11	2020
295303	PENDIDIKAN	2.299105e+09	2020

[22298 rows x 3 columns]

10 Penghitungan Nilai Anggaran Pertahun di sektor Pendidikan

```
[ ]: # Menghitung total nilai anggaran per tahun
total_anggaran_per_tahun = data_pendidikan.
    ↳groupby('periode_data')['nilai_anggaran'].sum().reset_index()

# Mengganti nama kolom untuk kemudahan pemahaman
total_anggaran_per_tahun.rename(columns={'nilai_anggaran': '
    ↳total_nilai_anggaran'}, inplace=True)

# Fungsi untuk mengonversi angka ke format Rupiah
def format_rupiah(value):
    if value >= 1e12:
        return f"Rp {value / 1e12:.2f} Triliun"
    elif value >= 1e9:
        return f"Rp {value / 1e9:.2f} Miliar"
    elif value >= 1e6:
        return f"Rp {value / 1e6:.2f} Juta"
    else:
        return f"Rp {value:.2f}"

# Terapkan format Rupiah ke kolom total_nilai_anggaran
total_anggaran_per_tahun['total_nilai_anggaran'] =
    ↳total_anggaran_per_tahun['total_nilai_anggaran'].apply(format_rupiah)

# Menampilkan hasilnya
print("Total Nilai Anggaran Pendidikan per Tahun:")
print(total_anggaran_per_tahun)
```

Total Nilai Anggaran Pendidikan per Tahun:

	periode_data	total_nilai_anggaran
0	2014	Rp 33.10 Miliar
1	2015	Rp 3.90 Triliun
2	2016	Rp 4.46 Triliun
3	2017	Rp 4.09 Triliun

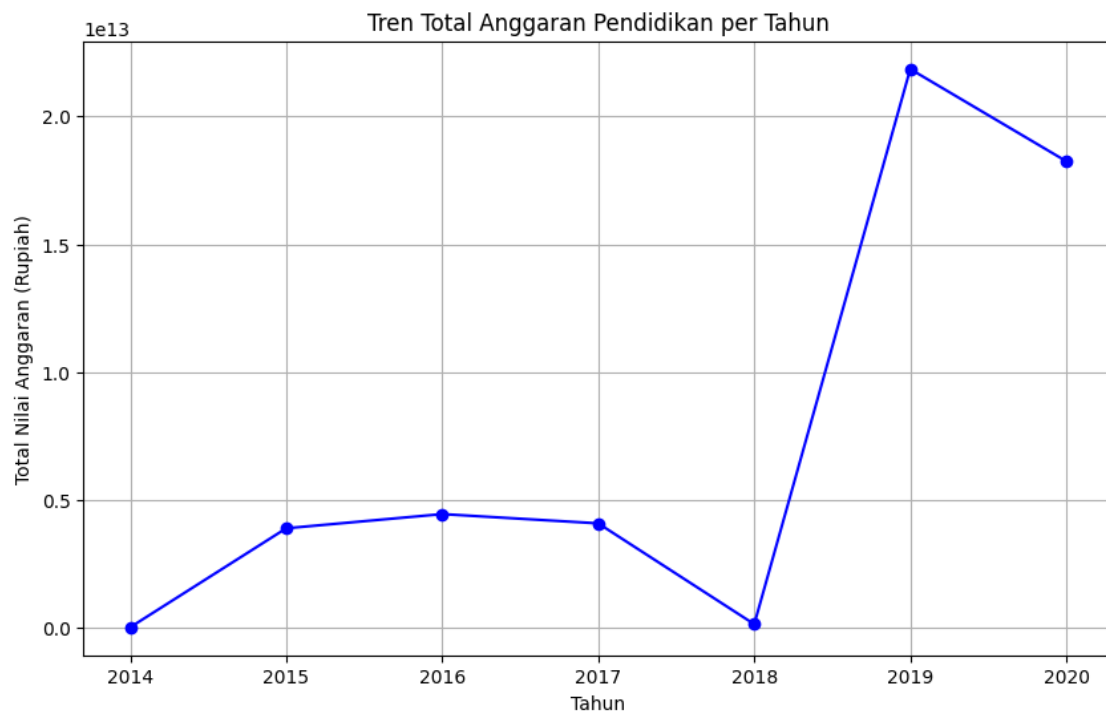
4	2018	Rp 153.36 Miliar
5	2019	Rp 21.85 Triliun
6	2020	Rp 18.25 Triliun

11 Membuat Grafik Tren Total Anggaran Pendidikan per Tahun

```
[ ]: # Menghitung total nilai anggaran per tahun
total_anggaran_per_tahun = data_pendidikan.
    ↳groupby('periode_data')['nilai_anggaran'].sum().reset_index()

# Membuat plot time series
plt.figure(figsize=(10, 6))
plt.plot(total_anggaran_per_tahun['periode_data'],
    ↳total_anggaran_per_tahun['nilai_anggaran'], marker='o', color='b',
    ↳linestyle='-')
plt.title("Tren Total Anggaran Pendidikan per Tahun")
plt.xlabel("Tahun")
plt.ylabel("Total Nilai Anggaran (Rupiah)")
plt.xticks(total_anggaran_per_tahun['periode_data']) # Menampilkan setiap
    ↳tahun sebagai label pada sumbu x
plt.grid(True)

# Tampilkan grafik
plt.show()
```



Pada Tahun 2019, mengalami kenaikan anggaran yang sangat signifikan dikarenakan pada tahun tersebut DPRD DKI Jakarta menggenjot dana alokasi untuk sarana dan prasarana.

Sumber: DPRD DKI Jakarta. (Tahun). Peningkatan Alokasi Anggaran Pendidikan Sudah Seharusnya Dilakukan. Diakses dari <https://dprd-dkijakartaprov.go.id/peningkatan-alokasi-anggaran-pendidikan-sudah-seharusnya-dilakukan/>.

Filtering data untuk sektor Kesehatan pada dataset APBD 2019

```
[ ]: # Memfilter data khusus untuk "KESEHATAN" pada dataset APBD 2019
data_kesehatan_2019 = apbd2019.loc[apbd2019['nama_urusan_program'] == 'KESEHATAN', ['nama_urusan_program', 'nilai_anggaran', 'periode_data']]
data_kesehatan_2019.reset_index(drop=True, inplace=True)

# Menampilkan hasilnya
print(data_kesehatan_2019.head())
```

	nama_urusan_program	nilai_anggaran	periode_data
0	KESEHATAN	349000	2019
1	KESEHATAN	26902095	2019
2	KESEHATAN	296450000	2019
3	KESEHATAN	21385000	2019
4	KESEHATAN	103890600	2019

Filtering data untuk sektor Kewilayahan pada dataset APBD 2019

```
[ ]: # Memfilter data khusus untuk "KESEHATAN" pada dataset APBD 2019
data_kesehatan_2019 = apbd2019.loc[apbd2019['nama_urusan_program'] == 'KEWILAYAHAN', ['nama_urusan_program', 'nilai_anggaran', 'periode_data']]
data_kesehatan_2019.reset_index(drop=True, inplace=True)

# Menampilkan hasilnya
print(data_kesehatan_2019.head())
```

	nama_urusan_program	nilai_anggaran	periode_data
0	KEWILAYAHAN	10700250	2019
1	KEWILAYAHAN	111888000	2019
2	KEWILAYAHAN	42200413	2019
3	KEWILAYAHAN	2053084	2019
4	KEWILAYAHAN	174000	2019

Menghitung Total Nilai Anggaran Kesehatan pada dataset APBD 2019 dalam format Rupiah

```
[ ]: # Memfilter data untuk sektor kesehatan pada dataset APBD 2019
data_kesehatan_2019 = apbd2019.loc[apbd2019['nama_urusan_program'] == 'KESEHATAN', ['nilai_anggaran']]
```

```

# Menghapus koma dari nilai_anggaran dan mengubah tipe data menjadi integer
data_kesehatan_2019['nilai_anggaran'] = (
    data_kesehatan_2019['nilai_anggaran']
    .replace(',', '', regex=True) # Menghapus koma
    .astype(int) # Mengonversi ke integer
)

# Menghitung total nilai anggaran
total_nilai_anggaran_kesehatan_2019 = data_kesehatan_2019['nilai_anggaran'].
    ↪sum()

# Fungsi untuk mengonversi angka ke format Rupiah
def format_rupiah(value):
    if value >= 1e12:
        return f"Rp {value / 1e12:.2f} Triliun"
    elif value >= 1e9:
        return f"Rp {value / 1e9:.2f} Miliar"
    elif value >= 1e6:
        return f"Rp {value / 1e6:.2f} Juta"
    else:
        return f"Rp {value:.2f}"

# Menggunakan fungsi untuk mengformat total nilai anggaran
total_nilai_anggaran_kesehatan_2019_formatted = ↪
    ↪format_rupiah(total_nilai_anggaran_kesehatan_2019)

# Menampilkan total nilai anggaran dalam format Rupiah
print(f"Total Nilai Anggaran Kesehatan DKI Jakarta tahun 2019:↪
    ↪{total_nilai_anggaran_kesehatan_2019_formatted}")

```

Total Nilai Anggaran Kesehatan DKI Jakarta tahun 2019: Rp 12.12 Triliun

Menghitung Total Nilai Anggaran Kesehatan pada dataset APBD 2019 dalam format Rupiah

```

[ ]: # Memfilter data untuk sektor kesehatan pada dataset APBD 2019
data_kewilayahan_2019 = apbd2019.loc[apbd2019['nama_urusan_program'] ==↪
    ↪'KEWILAYAHAN', ['nilai_anggaran']]

# Menghapus koma dari nilai_anggaran dan mengubah tipe data menjadi integer
data_kewilayahan_2019['nilai_anggaran'] = (
    data_kewilayahan_2019['nilai_anggaran']
    .replace(',', '', regex=True) # Menghapus koma
    .astype(int) # Mengonversi ke integer
)

# Menghitung total nilai anggaran

```

```

total_nilai_anggaran_kewilayahan_2019 = data_kewilayahan_2019['nilai_anggaran'].
↳sum()

# Fungsi untuk mengonversi angka ke format Rupiah
def format_rupiah(value):
    if value >= 1e12:
        return f"Rp {value / 1e12:.2f} Triliun"
    elif value >= 1e9:
        return f"Rp {value / 1e9:.2f} Miliar"
    elif value >= 1e6:
        return f"Rp {value / 1e6:.2f} Juta"
    else:
        return f"Rp {value:.2f}"

# Menggunakan fungsi untuk mengformat total nilai anggaran
total_nilai_anggaran_kewilayahan_2019_formatted =
↳format_rupiah(total_nilai_anggaran_kewilayahan_2019)

# Menampilkan total nilai anggaran dalam format Rupiah
print(f"Total Nilai Anggaran Kewilayahan DKI Jakarta tahun 2019:
↳{total_nilai_anggaran_kewilayahan_2019_formatted}")

```

Total Nilai Anggaran Kewilayahan DKI Jakarta tahun 2019: Rp 4.05 Triliun

12 Membuat Grafik untuk Perbandingan Total Nilai Anggaran pada sektor Pendidikan, Kesehatan, dan Kewilayahan pada Tahun 2019

```

[ ]: # Menghitung total nilai anggaran untuk sektor pendidikan di tahun 2019
data_pendidikan_2019 = apbd2019.loc[apbd2019['nama_urusan_program'] ==
↳'PENDIDIKAN', ['nilai_anggaran']]
data_pendidikan_2019['nilai_anggaran'] = (
    data_pendidikan_2019['nilai_anggaran']
    .replace(',', '', regex=True) # Menghapus koma
    .astype(int) # Mengonversi ke integer
)
total_nilai_anggaran_pendidikan_2019 = data_pendidikan_2019['nilai_anggaran'].
↳sum()

# Menghitung total nilai anggaran untuk sektor kesehatan di tahun 2019
data_kesehatan_2019 = apbd2019.loc[apbd2019['nama_urusan_program'] ==
↳'KESEHATAN', ['nilai_anggaran']]
data_kesehatan_2019['nilai_anggaran'] = (
    data_kesehatan_2019['nilai_anggaran']
    .replace(',', '', regex=True) # Menghapus koma
    .astype(int) # Mengonversi ke integer
)

```

```

)
total_nilai_anggaran_kesehatan_2019 = data_kesehatan_2019['nilai_anggaran'].
↳sum()

# Menghitung total nilai anggaran untuk sektor kewilayahan di tahun 2019
data_kewilayahan_2019 = apbd2019.loc[apbd2019['nama_urusan_program'] == '
↳'KEWILAYAHAN', ['nilai_anggaran']]
data_kewilayahan_2019['nilai_anggaran'] = (
    data_kewilayahan_2019['nilai_anggaran']
    .replace(',', '', regex=True) # Menghapus koma
    .astype(int) # Mengonversi ke integer
)
total_nilai_anggaran_kewilayahan_2019 = data_kewilayahan_2019['nilai_anggaran'].
↳sum()

# Menyusun data untuk grafik
sektor = ['Pendidikan', 'Kesehatan', 'Kewilayahan']
total_nilai_anggaran = [
    total_nilai_anggaran_pendidikan_2019,
    total_nilai_anggaran_kesehatan_2019,
    total_nilai_anggaran_kewilayahan_2019
]

# Membuat grafik batang
plt.figure(figsize=(10, 6))
bars = plt.bar(sektor, total_nilai_anggaran, color=['#4CAF50', '#2196F3', '
↳'#FF9800'], edgecolor='black')

# Menambahkan detail pada grafik
plt.title('Perbandingan Total Nilai Anggaran Pendidikan, Kesehatan, dan
↳Kewilayahan DKI Jakarta Tahun 2019', fontsize=14, fontweight='bold')
plt.xlabel('Sektor', fontsize=12)
plt.ylabel('Total Nilai Anggaran (Rupiah)', fontsize=12)
plt.xticks(sektor, fontsize=12)
plt.yticks(fontsize=12)

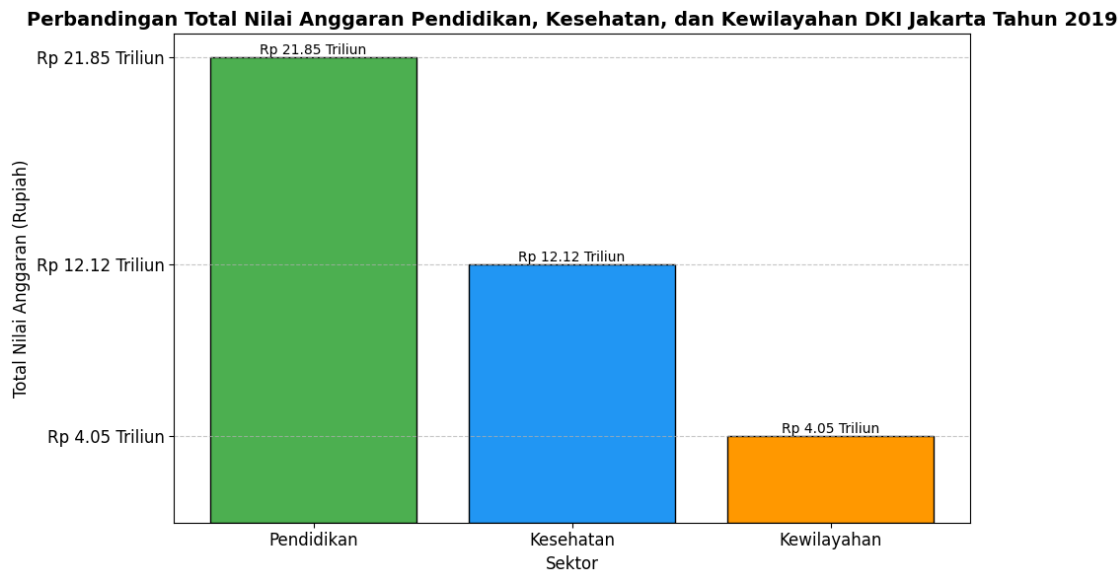
# Menambahkan format Rupiah ke sumbu Y
def format_rupiah(value):
    if value >= 1e12:
        return f"Rp {value / 1e12:.2f} Triliun"
    elif value >= 1e9:
        return f"Rp {value / 1e9:.2f} Miliar"
    elif value >= 1e6:
        return f"Rp {value / 1e6:.2f} Juta"
    else:
        return f"Rp {value:.2f}"

```

```
# Mengubah label sumbu Y menjadi format Rupiah
plt.yticks(total_nilai_anggaran, [format_rupiah(x) for x in
    ↪total_nilai_anggaran])

# Menambahkan anotasi untuk nilai anggaran di atas setiap batang
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, format_rupiah(yval),
    ↪ha='center', va='bottom', fontsize=10)

# Menambahkan grid untuk memudahkan pembacaan
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout() # Mengatur layout agar lebih rapi
plt.show()
```



Pada Tahun 2019, Sektor Pendidikan memiliki anggaran yang lebih tinggi dibandingkan dengan sektor Kesehatan dan Kewilayahan dikarenakan Pendanaan pendidikan minimal 20 persen diatur dalam Undang-Undang Dasar 1945 Pasal 31 Ayat 4, yang menyebutkan anggaran pendidikan harus setidaknya 20% dari APBN dan APBD.

Sumber: DPRD DKI Jakarta. (Tahun). Peningkatan Alokasi Anggaran Pendidikan Sudah Seharusnya Dilakukan. Diakses dari <https://dprd-dkijakartaprof.go.id/peningkatan-alokasi-anggaran-pendidikan-sudah-seharusnya-dilakukan/>.

```
[ ]: # Kamus singkatan untuk nama sektor
singkatan_nama = {
    "OTONOMI DAERAH, PEMERINTAHAN UMUM, ADMINISTRASI KEUANGAN DAERAH, PERANGKAT_
    ↪DAERAH, KEPEGAWAIAN, DAN PERSANDIAN": "OTONOMI & ADMIN",
```

```

        "PEKERJAAN UMUM DAN PENATAAN RUANG": "PEKERJAAN UMUM",
        "FUNGSI LAIN SESUAI DENGAN KETENTUAN PERATURAN PERUNDANG-UNDANGAN": "FUNGSI_
↳LAIN",
        "PEMBERDAYAAN PEREMPUAN DAN PERLINDUNGAN ANAK": "PEMBERDAYAAN ANAK",
        "PENELITIAN DAN PENGEMBANGAN": "PENELITIAN & PENGEMBANGAN",
        "KEARSIPAN": "ARSIP",
        "PERSANDIAN": "SANDI",
        "STATISTIK": "STATISTIK",
        "KESEHATAN": "KESEHATAN",
        "PENDIDIKAN": "PENDIDIKAN"
    }

    # Terapkan singkatan pada nama sektor
    sektor_terbesar['nama_urusan_program'] = sektor_terbesar['nama_urusan_program'].
↳replace(singkatan_nama)
    sektor_terkecil['nama_urusan_program'] = sektor_terkecil['nama_urusan_program'].
↳replace(singkatan_nama)

    # Fungsi untuk memformat angka ke triliun
    def format_triliun(x, pos):
        return f'{x * 1e-12:.1f}T'

    # Fungsi untuk memformat angka ke miliar
    def format_miliar(x, pos):
        return f'{x * 1e-9:.1f}M'

    # Langkah 4: Buat subplot
    fig, axs = plt.subplots(1, 2, figsize=(14, 6))

    # Grafik untuk sektor terbesar
    axs[0].barh(sektor_terbesar['nama_urusan_program'],_
↳sektor_terbesar['nilai_anggaran'], color='blue')
    axs[0].set_title('5 Sektor Terbesar Anggaran APBD 2019')
    axs[0].set_xlabel('Total Anggaran (Triliun Rupiah)')
    axs[0].set_ylabel('Sektor')
    axs[0].xaxis.set_major_formatter(FuncFormatter(format_triliun)) # Terapkan_
↳format ke sumbu x sektor terbesar

    # Grafik untuk sektor terkecil
    axs[1].barh(sektor_terkecil['nama_urusan_program'],_
↳sektor_terkecil['nilai_anggaran'], color='orange')
    axs[1].set_title('5 Sektor Terkecil Anggaran APBD 2019')
    axs[1].set_xlabel('Total Anggaran (Miliar Rupiah)')
    axs[1].set_ylabel('Sektor')
    axs[1].xaxis.set_major_formatter(FuncFormatter(format_miliar)) # Terapkan_
↳format ke sumbu x sektor terkecil

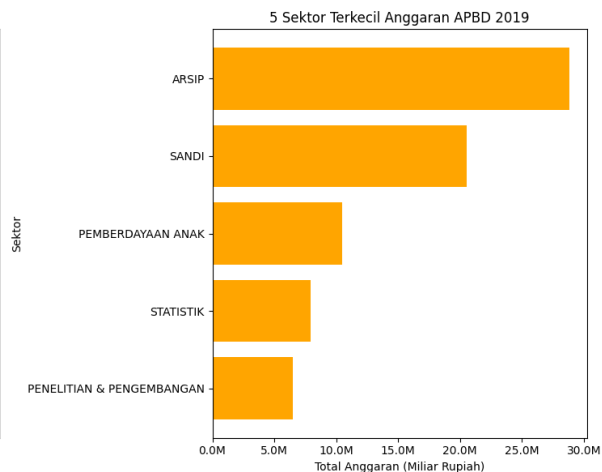
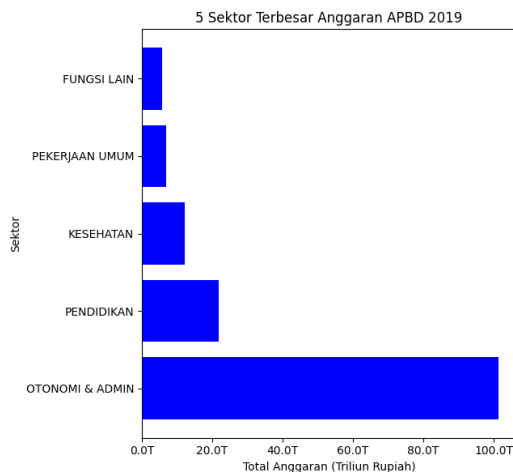
```

```
# Tampilkan 5 sektor terkecil
print(sektor_terkecil.head())
print(sektor_terbesar.head())

# Atur tata letak
plt.tight_layout()
plt.show()
```

	nama_urusan_program	nilai_anggaran
33	PENELITIAN & PENGEMBANGAN	6525862955
47	STATISTIK	7961050583
28	PEMBERDAYAAN ANAK	10499436755
41	SANDI	20540978744
4	ARSIP	28816010268

	nama_urusan_program	nilai_anggaran
22	OTONOMI & ADMIN	101385705083634
32	PENDIDIKAN	21851026087092
13	KESEHATAN	12121957907890
26	PEKERJAAN UMUM	6873531471428
2	FUNGSI LAIN	5834619322114



Data diatas menunjukkan 5 Sektor Penerima Anggaran Terbesar dan Terkecil

Kesimpulan

Analisis ini menunjukkan fluktuasi yang signifikan dalam anggaran pendidikan DKI Jakarta dari tahun 2014 hingga 2020. Pada tahun 2014, anggaran pendidikan mencapai IDR 33,10 miliar, angka terendah selama periode ini. Namun, pada tahun 2017, anggaran meningkat tajam menjadi IDR 4,09 triliun, mencerminkan perhatian yang lebih besar terhadap sektor pendidikan. Sayangnya, pada tahun 2018, anggaran pendidikan mengalami penurunan drastis menjadi IDR 153,36 miliar, menandakan adanya kendala dalam pengalokasian dana.

Pada tahun 2019, terjadi lonjakan luar biasa hingga IDR 21,85 triliun, yang menunjukkan komitmen pemerintah daerah untuk meningkatkan kualitas pendidikan. Lonjakan ini dapat dilihat sebagai respons terhadap kebutuhan mendesak dalam pengembangan infrastruktur pendidikan dan peningkatan kualitas layanan pendidikan.

Secara keseluruhan, meskipun terdapat beberapa tahun dengan pengurangan anggaran, tren peningkatan yang signifikan pada tahun-tahun tertentu menunjukkan bahwa anggaran pendidikan tetap menjadi prioritas dalam kebijakan APBD DKI Jakarta. Penelitian lebih lanjut diperlukan untuk menganalisis faktor-faktor yang memengaruhi perubahan ini dan untuk mengidentifikasi dampak nyata dari pengeluaran tersebut terhadap kualitas pendidikan di DKI Jakarta.