

**LAPORAN DATA ANALISIS PPH BADAN MENGGUNAKAN
BIG QUERY DAN GOOGLE COLAB**
Disusun Untuk Memenuhi Tugas Akhir
Mata Kuliah Pengkodean dan Pemrograman



Disusun Oleh:
Fathinatul Hasanah
12030123140294

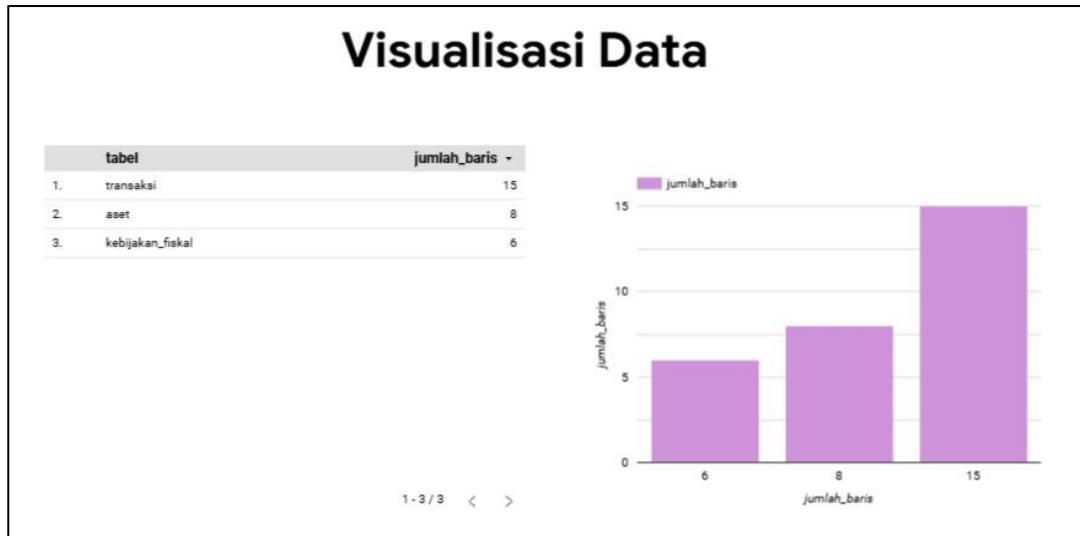
Dosen Pengampu:
Dr. Totok Dewayanto. S.E., M.Si., Akt.

**PROGRAM STUDI AKUNTANSI
FAKULTAS EKONOMIKA DAN BISNIS
UNIVERSITAS DIPONEGORO
2025**

Visualisasi Data

1. BigQuery

```
-- Menghitung jumlah baris untuk setiap tabel
WITH table_counts AS (
    SELECT 'transaksi' AS tabel, COUNT(*) AS jumlah_baris FROM `utilitarian-web-459016-v4.PPh_Badan.transaksi_keuangan`
    UNION ALL
    SELECT 'aset' AS tabel, COUNT(*) AS jumlah_baris FROM `utilitarian-web-459016-v4.PPh_Badan.aset_tetap`
    UNION ALL
    SELECT 'kebijakan_fiskal' AS tabel, COUNT(*) AS jumlah_baris FROM `utilitarian-web-459016-v4.PPh_Badan.kebijakan_fiskal`
)
SELECT
    tabel,
    jumlah_baris
FROM table_counts
ORDER BY tabel;
```



2. Google Colab

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

# Data jumlah baris dari masing-masing tabel
tables = ['transaksi', 'aset', 'kebijakan_fiskal']
counts = [15, 8, 6]

# Membuat figure dan axes untuk 3D plot
fig = plt.figure(figsize=(10, 8))
```

```
ax = fig.add_subplot(111, projection='3d')

# Posisi batang pada sumbu X dan Y
xpos = np.arange(len(tables))
ypos = np.zeros_like(xpos) # Gunakan satu nilai untuk sumbu Y agar
bar terlihat berdampingan
zpos = np.zeros_like(xpos)

# Lebar dan kedalaman batang
dx = 0.8 * np.ones_like(zpos)
dy = 0.5 * np.ones_like(zpos) # Kedalaman batang
dz = counts # Tinggi batang (jumlah baris)

# Warna batang
colors = ['skyblue', 'lightcoral', 'lightgreen']
bar_colors = [colors[i % len(colors)] for i in range(len(tables))]

# Membuat 3D bar chart
ax.bar3d(xpos, ypos, zpos, dx, dy, dz, color=bar_colors, alpha=0.8)

# Atur label sumbu
ax.set_xlabel('Tabel', fontsize=12)
ax.set_ylabel('', fontsize=12) # Kosongkan label Y karena tidak
digunakan secara fungsional
ax.set_zlabel('Jumlah Baris', fontsize=12)

# Atur tick dan label sumbu X (Tabel)
ax.set_xticks(xpos + dx[0]/2) # Posisikan tick di tengah batang
ax.set_xticklabels(tables)

# Sembunyikan tick sumbu Y
ax.set_yticks([])

# Judul plot
plt.title('VISUALISASI DATA', fontsize=16) # Mengubah judul plot

# Sesuaikan sudut pandang
ax.view_init(elev=20, azim=-45)

plt.show()
```

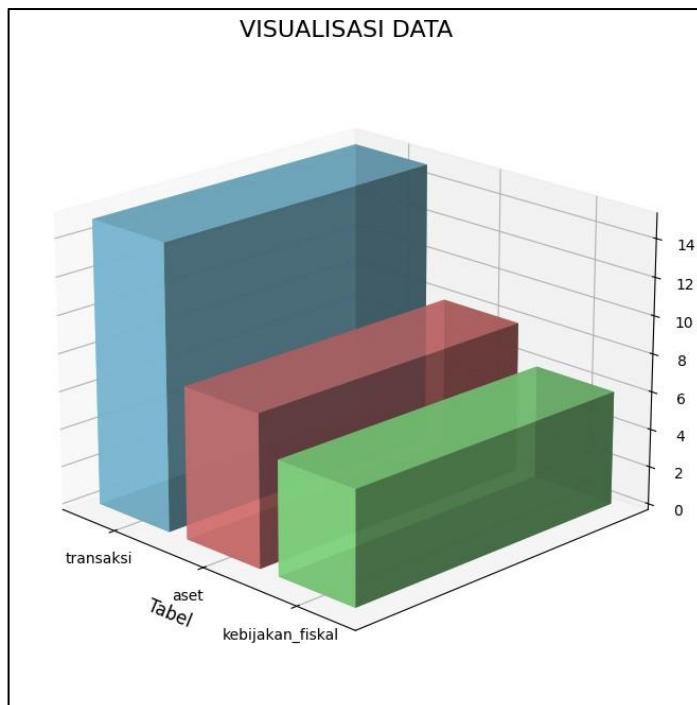


table	jumlah baris
transaksi	15
aset	8
kebijakan_fiskal	6

Analisis:

1. Visualisasi Data dari BigQuery

Visualisasi pada bagian atas merupakan output dari BigQuery, yang menampilkan jumlah baris pada tiap tabel dalam dua format, yaitu tabel data dan grafik batang 2D.

- Tabel Data menampilkan tiga baris:
 - transaksi : 15 baris
 - asset : 8 baris
 - kebijakan_fiskal : 6 baris
- Grafik Batang 2D berwarna ungu menunjukkan representasi visual dari tabel tersebut secara vertikal, dengan sumbu X adalah nilai jumlah baris (6, 8, 15) dan sumbu Y adalah label jumlah_baris. Setiap batang menunjukkan ketinggian yang proporsional dengan jumlah baris per tabel.

Kelebihan visualisasi BigQuery:

- Sangat jelas dan mudah dibaca.
- Data numerik langsung ditampilkan dalam tabel.
- Grafik 2D sederhana dan efektif dalam memperlihatkan perbandingan antar tabel.

2. Visualisasi Data dari Google Colab

Visualisasi kedua (tengah gambar) dibuat menggunakan Google Colab, yang menyajikan data yang sama dalam bentuk grafik batang 3D.

- Tiga batang vertikal 3D:

- Warna biru untuk transaksi, dengan ketinggian menunjukkan 15 baris.
- Warna merah untuk aset, menunjukkan 8 baris.
- Warna hijau untuk kebijakan_fiskal, menunjukkan 6 baris.
- Setiap batang berada pada sumbu horizontal yang mewakili nama tabel, dengan sumbu vertikal menunjukkan jumlah baris.

Kelbihan visualisasi Google Colab:

- Lebih menarik secara visual dan interaktif.
- Membantu pengguna memahami distribusi data secara spasial.

Namun kelemahannya:

- Grafik 3D bisa membuat pembacaan angka kurang presisi karena perspektif dan sudut pandang bisa menipu proporsi sebenarnya.
- Tidak menampilkan angka secara langsung (tidak ada label data di batang), sehingga pengguna harus menebak dari tinggi batang.

Kesimpulan:

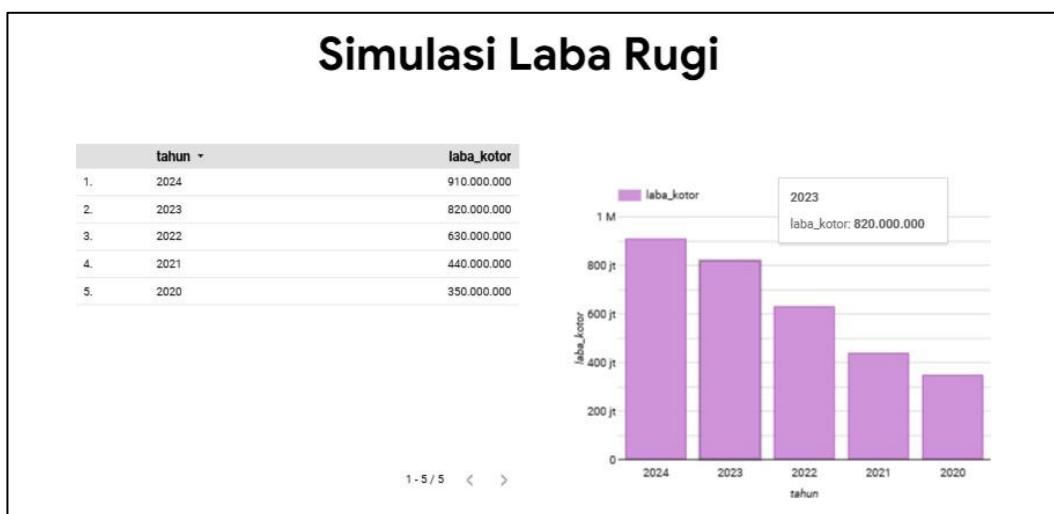
Kedua visualisasi menyampaikan informasi yang sama, yaitu jumlah baris pada masing-masing tabel (transaksi, aset, kebijakan_fiskal). Namun, visualisasi dari BigQuery lebih unggul dari segi kejelasan dan keterbacaan, sangat cocok untuk analisis cepat dan akurat. Sebaliknya, visualisasi dari Google Colab lebih estetis dan cocok untuk presentasi visual, tetapi kurang ideal untuk analisis numerik cepat karena perspektif 3D bisa mengganggu persepsi akurat terhadap data.

Skenario Simulasi Laba Rugi

1. BigQuery

```
-- A. Simulasi Laba/Rugi Skenario Optimistis
-- Menghitung laba kotor (pendapatan - (beban_operasional + penyusutan)) untuk
skenario Optimistis

SELECT
    tahun,
    SUM(pendapatan) - SUM(beban_operasional + penyusutan) AS laba_kotor
FROM `utilitarian-web-459016-v4.PPh_Badan.transaksi_keuangan`
WHERE skenario = 'Optimistis'
GROUP BY tahun
ORDER BY tahun;
```



2. Google Colab

```
# Visualisasi 3D Bar Chart untuk semua skenario laba kotor
fig = plt.figure(figsize=(14, 10))
ax = fig.add_subplot(111, projection='3d')

# Dapatkan daftar tahun dan skenario unik
years = laba_kotor_per_tahun_skenario['tahun'].unique()
scenarios = laba_kotor_per_tahun_skenario['skenario'].unique()

# Buat posisi untuk batang pada sumbu x (tahun) dan y (skenario)
xpos, ypos = np.meshgrid(np.arange(len(years)),
np.arange(len(scenarios)))
xpos = xpos.flatten()
ypos = ypos.flatten()
zpos = np.zeros(len(xpos))

# Lebar dan kedalaman batang
# dx dan dy harus sesuai dengan jumlah total batang (tahun * skenario)
dx = 0.5 * np.ones_like(xpos)
dy = 0.5 * np.ones_like(ypos)

dz = laba_kotor_per_tahun_skenario['laba_kotor'].values / 1e9 # Tinggi batang (laba kotor dalam Miliar)

# Mapping warna untuk setiap skenario
colors = {'Optimistis': 'dodgerblue', 'Moderat': 'limegreen',
'Pesimistis': 'salmon'}
# Buat daftar warna untuk setiap batang
scenario_colors = [colors[skenario] for skenario in
laba_kotor_per_tahun_skenario['skenario']]

# Buat 3D bar chart
ax.bar3d(xpos, ypos, zpos, dx, dy, dz, color=scenario_colors,
alpha=0.8)

# Atur label sumbu
ax.set_xlabel('Tahun', fontsize=12)
ax.set_ylabel('Skenario', fontsize=12)
ax.set_zlabel('Laba Kotor (Miliar Rupiah)', fontsize=12)

# Atur tick dan label sumbu. Ticks harus berada di tengah kelompok batang untuk setiap tahun dan skenario.
# Untuk sumbu X (Tahun), setiap kelompok batang (Optimistis, Moderat, Pesimistis) untuk satu tahun berada pada posisi xpos tahun tersebut.
# Kita perlu menghitung posisi tengah untuk setiap tahun pada sumbu X
x_tick_positions = np.arange(len(years)) + dx[0]/2 # Menggunakan satu nilai dx karena semua batang memiliki lebar yang sama
ax.set_xticks(x_tick_positions)
```

```

ax.set_xticklabels(years)

# Untuk sumbu Y (Skenario), setiap kelompok batang untuk satu skenario berada pada posisi ypos skenario tersebut.
y_tick_positions = np.arange(len(scenarios)) + dy[0]/2 # Menggunakan satu nilai dy
ax.set_yticks(y_tick_positions)
ax.set_yticklabels(scenarios)

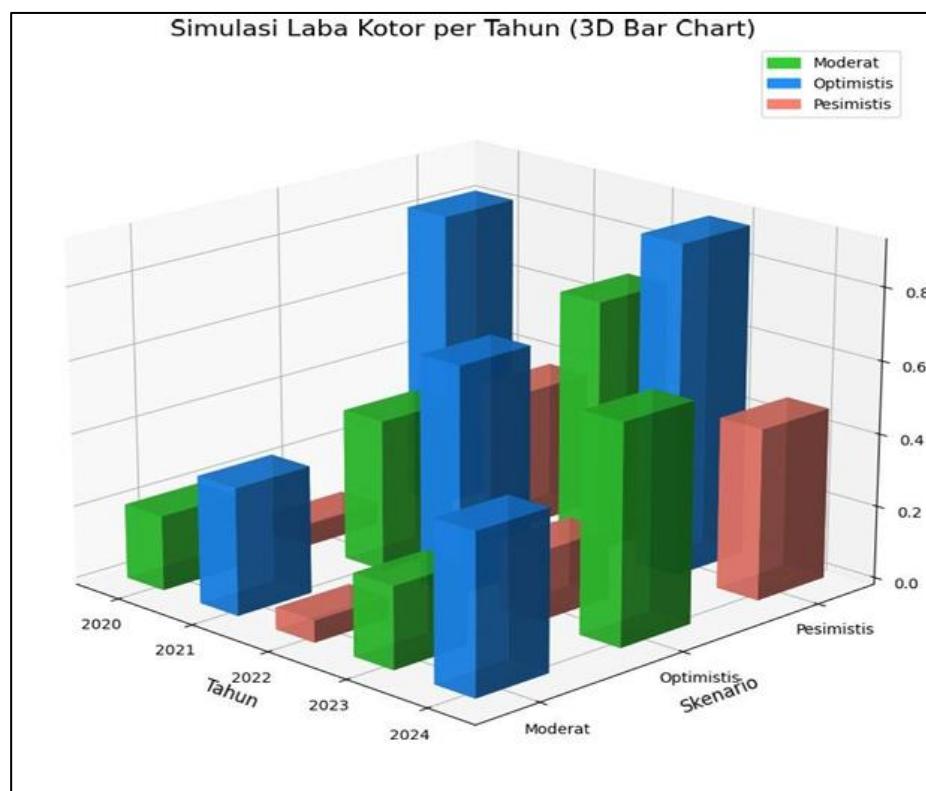
# Judul plot
ax.set_title('Simulasi Laba Kotor per Tahun (3D Bar Chart)', fontsize=16)

# Tambahkan legenda manual karena bar3d tidak otomatis membuat legenda
import matplotlib.patches as mpatches
legend_patches = [mpatches.Patch(color=colors[skenario], label=skenario) for skenario in scenarios]
ax.legend(handles=legend_patches)

# Sesuaikan sudut pandang
ax.view_init(elev=20, azim=-45)

plt.show()

```



tahun	skenario	laba_kotor
2020	Moderat	205000000
2020	Optimistis	350000000
2020	Pesimistik	60000000
2021	Moderat	225000000
2021	Optimistis	440000000
2021	Pesimistik	60000000
2022	Moderat	415000000
2022	Optimistis	630000000
2022	Pesimistik	190000000
2023	Moderat	605000000
2023	Optimistis	820000000
2023	Pesimistik	380000000
2024	Moderat	695000000
2024	Optimistis	910000000
2024	Pesimistik	470000000

Analisis:

1. Simulasi Laba Rugi BigQuery

Visualisasi dari BigQuery menampilkan tabel laba kotor per tahun dari 2020 hingga 2024. Data ditampilkan dalam bentuk tabel dan diagram batang vertikal berwarna ungu.

- Isi Tabel:
 - Tahun 2020 hingga 2024.
 - Nilai laba kotor meningkat secara bertahap setiap tahunnya.
 - Contoh: tahun 2020 sebesar 350 juta, dan naik hingga 910 juta di tahun 2024.
- Isi Diagram:
 - Setiap batang mewakili laba kotor pada satu tahun.
 - Tinggi batang menunjukkan besarnya laba, yang meningkat dari kiri (2020) ke kanan (2024).
 - Diagram ini membantu melihat tren pertumbuhan laba kotor secara visual dengan cepat.

Kesimpulan dari visualisasi BigQuery: data menunjukkan bahwa perusahaan mengalami pertumbuhan laba kotor secara konsisten dari tahun ke tahun tanpa adanya fluktuasi atau skenario alternatif.

2. Simulasi Laba Rugi Google Colab

Visualisasi dari Google Colab menampilkan simulasi laba kotor per tahun berdasarkan tiga skenario: Moderat, Optimistis, dan Pesimistik dalam bentuk diagram batang 3 dimensi.

- Isi Diagram:
 - Terdapat tiga batang pada setiap tahun (2020–2024), masing-masing berwarna:
 - Hijau untuk Moderat
 - Biru untuk Optimistis
 - Merah untuk Pesimistik
 - Tinggi batang bervariasi tergantung skenario, menunjukkan bahwa untuk tahun yang sama, nilai laba bisa berbeda tergantung kondisi yang terjadi.
- Tren Data:

- Skenario optimistis selalu memiliki nilai laba tertinggi.
- Skenario pesimistik menampilkan proyeksi laba terendah.
- Tahun 2024 menunjukkan laba tertinggi pada semua skenario.
- Grafik ini memperlihatkan kemungkinan variasi hasil yang lebih realistik dan mencerminkan perencanaan yang berbasis risiko

Perbandingan:

- BigQuery hanya fokus pada satu skenario realisasi historis, sedangkan Google Colab menyajikan proyeksi multi-skenario.
- Visualisasi dari BigQuery cocok untuk melihat tren pertumbuhan aktual, sedangkan Google Colab lebih cocok untuk analisis strategis dan pengambilan keputusan.
- BigQuery menekankan data faktual, sementara Google Colab menekankan simulasi dan ketidakpastian masa depan.

Kesimpulan:

Visualisasi dari BigQuery memberikan gambaran tren sederhana dan mudah dipahami atas kenaikan laba kotor dari tahun ke tahun. Namun, ia hanya menampilkan satu kemungkinan jalur perkembangan, tanpa mempertimbangkan risiko atau ketidakpastian.

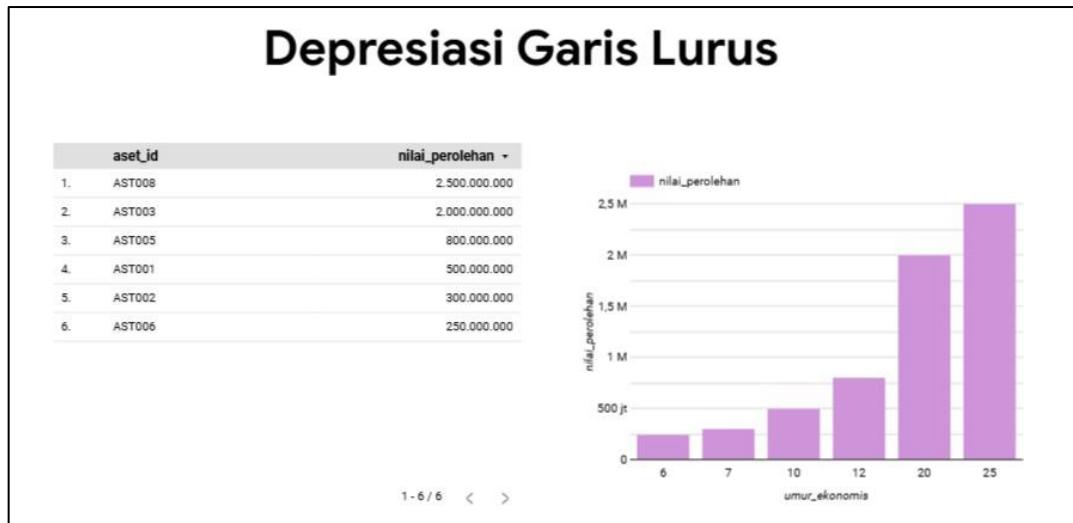
Sebaliknya, visualisasi dari Google Colab menyajikan analisis yang lebih mendalam dan realistik. Dengan adanya tiga skenario, perusahaan dapat memahami dampak dari berbagai situasi ekonomi terhadap laba kotor, sehingga sangat bermanfaat dalam pengambilan keputusan strategis.

Kesimpulannya, kedua visualisasi saling melengkapi: BigQuery cocok untuk evaluasi tren secara umum, sedangkan Google Colab lebih tepat untuk perencanaan dan analisis risiko keuangan.

Simulasi Depresiasi Tahunan (Metode Garis Lurus)

1. BigQuery

```
-- B1. Simulasi Depresiasi - Metode Garis Lurus
-- Menghitung depresiasi tahunan: nilai_perolehan / umur_ekonomis
SELECT
    aset_id,
    kategori,
    nilai_perolehan,
    umur_ekonomis,
    ROUND(nilai_perolehan / umur_ekonomis, 2) AS depresiasi_tahunan
FROM `utilitarian-web-459016-v4.PPh_Badan.asset_tetap`
WHERE metode = 'Garis_Lurus'
ORDER BY aset_id;
```



2. Google Colab

```
# Impor library yang diperlukan (pastikan semua sudah terinstal)
try:
    from google.cloud import bigquery
    import pandas as pd
    import matplotlib.pyplot as plt
    from mpl_toolkits.mplot3d import Axes3D
    from google.colab import files
    import os
except ImportError as e:
    print(f"Error: Library tidak terinstal. Silakan jalankan: !pip
install {str(e).split()[-1]}")
    !pip install google-cloud-bigquery pandas matplotlib
    from google.cloud import bigquery
    import pandas as pd
    import matplotlib.pyplot as plt
    from mpl_toolkits.mplot3d import Axes3D
    from google.colab import files
    import os

# Coba akses BigQuery atau gunakan data cadangan untuk metode Garis Lurus
try:
    client = bigquery.Client()
    # Query SQL untuk simulasi depresiasi metode Garis Lurus
    query = """
SELECT
    aset_id,
    kategori,
    nilai_perolehan,
    umur_ekonomis,
    ROUND(nilai_perolehan / umur_ekonomis, 2) AS depresiasi_tahunan
FROM `utilitarian-web-459016-v4.PPh_Badan.aset_tetap`
```

```

        WHERE metode = 'Garis_Lurus'
        ORDER BY aset_id;
"""

query_job = client.query(query)
df_garis_lurus = query_job.to_dataframe()
except Exception as e:
    print(f"Error saat menjalankan query BigQuery: {str(e)}")
    print("Menggunakan data cadangan dari CSV untuk metode Garis Lurus...")
    # Data cadangan berdasarkan aset_tetap.csv untuk metode Garis Lurus
data_garis_lurus = {
    'aset_id': ['AST001', 'AST002', 'AST003', 'AST005', 'AST006',
    'AST008'],
    'kategori': ['Mesin', 'Kendaraan', 'Bangunan', 'Mesin',
    'Kendaraan', 'Bangunan'],
    'nilai_perolehan': [500000000, 300000000, 2000000000,
    800000000, 250000000, 2500000000],
    'umur_ekonomis': [10, 7, 20, 12, 6, 25],
    'depresiasi_tahunan': [50000000.00, 42857142.86,
    100000000.00, 66666666.67, 41666666.67, 100000000.00]
}
df_garis_lurus = pd.DataFrame(data_garis_lurus)

# Cetak DataFrame untuk verifikasi
print("Data Depresiasi Metode Garis Lurus:")
print(df_garis_lurus)

# Persiapkan data untuk visualisasi 3D Bar Chart
if df_garis_lurus is not None and not df_garis_lurus.empty:
    aset_id = df_garis_lurus['aset_id']
    umur_ekonomis = df_garis_lurus['umur_ekonomis']
    depresiasi_tahunan = df_garis_lurus['depresiasi_tahunan']

    # Buat visualisasi 3D Bar Chart
    try:
        fig = plt.figure(figsize=(12, 8))
        ax = fig.add_subplot(111, projection='3d')

        # Buat posisi untuk batang
        xpos = np.arange(len(aset_id))
        ypos = umur_ekonomis.values
        zpos = np.zeros_like(xpos)

        # Lebar dan kedalaman batang
        dx = 0.8 * np.ones_like(zpos)
        dy = 1.0 * np.ones_like(zpos)

```

```

        dz = depresiasi_tahunan.values / 1e6 # Tinggi batang
(depresiasi dalam Juta Rupiah)

        # Warna batang (bisa disesuaikan, misalnya berdasarkan
kategori atau nilai depresiasi)
        colors = plt.cm.viridis(depresiasi_tahunan /
depresiasi_tahunan.max())

        # Buat 3D bar chart
        ax.bar3d(xpos, ypos, zpos, dx, dy, dz, color=colors,
alpha=0.8)

        # Atur label sumbu
        ax.set_xlabel('Aset ID', fontsize=12)
        ax.set_ylabel('Umur Ekonomis (Tahun)', fontsize=12)
        ax.set_zlabel('Depresiasi Tahunan (Juta Rupiah)', fontsize=12)

        # Atur tick dan label sumbu X (Aset ID)
        ax.set_xticks(xpos + dx[0]/2)
        ax.set_xticklabels(aset_id)

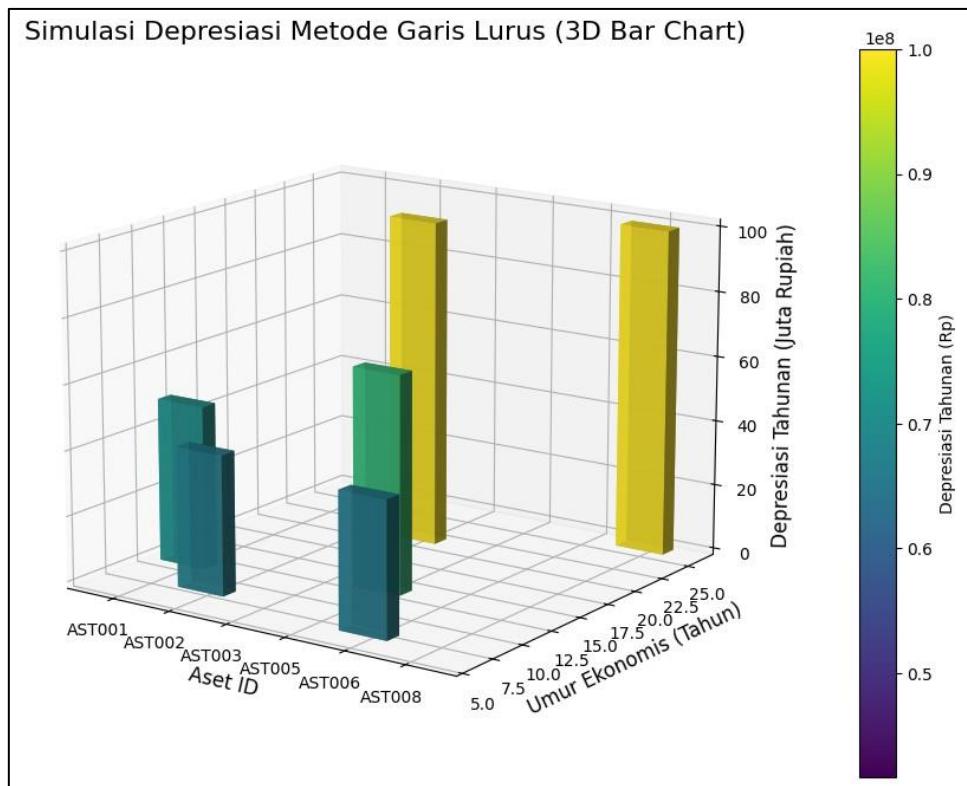
        # Judul plot
        ax.set_title('Simulasi Depresiasi Metode Garis Lurus (3D Bar
Chart)', fontsize=16)

        # Tambahkan colorbar jika menggunakan colormap
        cbar = fig.colorbar(plt.cm.ScalarMappable(cmap='viridis',
norm=plt.Normalize(vmin=depresiasi_tahunan.min(),
vmax=depresiasi_tahunan.max())), ax=ax, pad=0.1)
        cbar.set_label('Depresiasi Tahunan (Rp)')

        # Sesuaikan sudut pandang untuk kejelasan
        ax.view_init(elev=15, azim=-55) # Mengubah sudut pandang

        plt.show()
    except Exception as e:
        print(f"Error saat membuat visualisasi: {str(e)}")
        print("Pastikan matplotlib dan mpl_toolkits.mplot3d terinstal
dengan benar.")
    else:
        print("DataFrame untuk metode Garis Lurus kosong atau tidak
tersedia untuk visualisasi.")

```



aset_id	kategori	nilai_perolehan	umur_ekonomis	depresiasi_tahunan
AST001	Mesin	500000000	10	50000000.0
AST002	Kendaraan	300000000	7	42857142.86
AST003	Bangunan	2000000000	20	100000000.0
AST005	Mesin	800000000	12	66666666.67
AST006	Kendaraan	250000000	6	41666666.67
AST008	Bangunan	2500000000	25	100000000.0

Analisis:

1. Analisis Visualisasi dan Tabel dari BigQuery

BigQuery menampilkan data aset tetap dalam bentuk tabel sederhana dan grafik batang 2D. Data yang ditampilkan terdiri dari dua kolom:

- `aset_id`: Kode identifikasi aset
- `nilai_perolehan`: Harga perolehan awal aset dalam satuan rupiah

Grafik batang di sisi kanan menggambarkan hubungan antara umur ekonomis aset (sumbu X) dengan nilai perolehan aset (sumbu Y). Warna ungu digunakan untuk menandai kolom `nilai_perolehan`.

Interpretasi Visualisasi

Grafik memperlihatkan bahwa:

- Semakin tinggi nilai perolehan suatu aset, umumnya aset tersebut memiliki umur ekonomis yang lebih panjang.
- Misalnya, aset dengan nilai perolehan tertinggi (AST008 sebesar Rp2,5 miliar) memiliki umur ekonomis hingga 25 tahun.
- Sementara aset dengan nilai lebih rendah (seperti AST006 senilai Rp250 juta) memiliki umur ekonomis yang lebih pendek, sekitar 6 tahun.

Namun, belum ada informasi mengenai depresiasi tahunan, sehingga grafik ini hanya memberikan gambaran nilai awal aset dan umur manfaatnya.

Keterbatasan

- Tidak mencantumkan jenis/kategori aset.
- Tidak menunjukkan besarnya depresiasi tahunan.
- Belum menyajikan perhitungan finansial yang diperlukan untuk membuat keputusan manajerial atau akuntansi.

2. Analisis Visualisasi dan Tabel dari Google Colab

Data dari Google Colab menampilkan informasi yang lebih lengkap, meliputi:

- aset_id
- kategori aset (Mesin, Bangunan, Kendaraan)
- nilai_perolehan
- umur_ekonomis
- depresiasi_tahunan (yang dihitung menggunakan metode garis lurus)

Disertai dengan visualisasi 3D berbentuk batang yang menggambarkan hubungan antara:

- Aset ID (sumbu X)
- Umur ekonomis aset (sumbu Y)
- Depresiasi tahunan dalam rupiah (sumbu Z)

Interpretasi Visualisasi

- Aset dengan nilai tinggi dan umur panjang seperti Bangunan (AST003 dan AST008) memiliki depresiasi tahunan tertinggi, yaitu Rp100 juta per tahun.
- Aset berumur pendek seperti Kendaraan (AST006 dan AST002) memiliki depresiasi tahunan lebih rendah, meskipun nilai perolehannya juga rendah.
- Mesin (AST001 dan AST005) memiliki depresiasi menengah tergantung pada umur dan nilai aset.

Kelebihan

- Perhitungan depresiasi tahunan sudah tersedia, sehingga lebih praktis untuk digunakan dalam laporan keuangan.
- Informasi kategori aset mempermudah klasifikasi dan pengambilan keputusan.
- Visualisasi 3D lebih informatif dan menggambarkan keterkaitan antar variabel (nilai, umur, depresiasi).

Kesimpulan:

Berdasarkan analisis di atas, dapat disimpulkan bahwa:

1. Data dari BigQuery cocok digunakan pada tahap awal analisis, khususnya untuk:
 - Menyusun daftar aset
 - Mengelompokkan aset berdasarkan umur ekonomis
 - Melihat distribusi nilai perolehan aset
2. Data dari Google Colab lebih unggul dalam konteks pengambilan keputusan akuntansi dan keuangan, karena:
 - Sudah menghitung depresiasi tahunan berdasarkan metode garis lurus
 - Memuat informasi kategori aset, sehingga dapat digunakan untuk klasifikasi aset tetap
 - Menyediakan visualisasi 3D yang menggambarkan hubungan langsung antara umur, nilai, dan beban depresiasi

Oleh karena itu, jika menggunakan format data yang sama dan bertujuan untuk laporan keuangan, penganggaran, atau evaluasi aset tetap, Google Colab adalah

pilihan yang lebih tepat karena sudah menyertakan semua informasi penting yang dibutuhkan dalam analisis depresiasi.

Simulasi Depresiasi Tahunan (Metode Saldo Menurun)

1. BigQuery

```
-- B2. Simulasi Depresiasi - Metode Saldo Menurun
-- Menghitung depresiasi tahun pertama: nilai_perolehan * 0.25
-- Catatan: Kolom alternatif menggunakan 2x tarif garis lurus (2 / umur_ekonomis).
SELECT
    aset_id,
    kategori,
    nilai_perolehan,
    umur_ekonomis,
    ROUND(nilai_perolehan * 0.25, 2) AS depresiasi_tahun_pertama,
    ROUND(nilai_perolehan * (2.0 / umur_ekonomis), 2) AS
    depresiasi_tahun_pertama_alternatif
FROM `utilitarian-web-459016-v4.PPh_Badan.asset_tetap`
WHERE metode = 'Saldo_Menurun'
ORDER BY aset_id;
```



2. Google Colab

```
# Impor library yang diperlukan (pastikan semua sudah terinstal)
try:
    from google.cloud import bigquery
    import pandas as pd
    import matplotlib.pyplot as plt
    from mpl_toolkits.mplot3d import Axes3D
    from google.colab import files
    import os
except ImportError as e:
    print(f"Error: Library tidak terinstal. Silakan jalankan: !pip
install {str(e).split()[-1]}")
    !pip install google-cloud-bigquery pandas matplotlib
```

```

from google.cloud import bigquery
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from google.colab import files
import os

# Coba akses BigQuery atau gunakan data cadangan untuk metode Saldo Menurun
try:
    client = bigquery.Client()
    # Query SQL untuk simulasi depresiasi metode Saldo Menurun
    query = """
SELECT
    aset_id,
    kategori,
    nilai_perolehan,
    umur_ekonomis,
    ROUND(nilai_perolehan * 0.25, 2) AS depresiasi_tahun_pertama,
    ROUND(nilai_perolehan * (2.0 / umur_ekonomis), 2) AS
depresiasi_tahun_pertama_alternatif
FROM `utilitarian-web-459016-v4.PPh_Badan.aset_tetap`
WHERE metode = 'Saldo_Menurun'
ORDER BY aset_id;
"""

    query_job = client.query(query)
    df_saldo_menurun = query_job.to_dataframe()
except Exception as e:
    print(f"Error saat menjalankan query BigQuery: {str(e)}")
    print("Menggunakan data cadangan dari CSV untuk metode Saldo Menurun...")
    # Data cadangan berdasarkan aset_tetap.csv untuk metode Saldo Menurun
    data_saldo_menurun = {
        'aset_id': ['AST004', 'AST007'],
        'kategori': ['Peralatan', 'Peralatan'],
        'nilai_perolehan': [150000000, 100000000],
        'umur_ekonomis': [5, 4],
        'depresiasi_tahun_pertama': [37500000.00, 25000000.00],
        'depresiasi_tahun_pertama_alternatif': [60000000.00,
50000000.00]
    }
    df_saldo_menurun = pd.DataFrame(data_saldo_menurun)

# Cetak DataFrame untuk verifikasi
print("Data Depresiasi Metode Saldo Menurun:")
print(df_saldo_menurun)

```

```

# Persiapkan data untuk visualisasi 3D Bar Chart
if df_saldo_menurun is not None and not df_saldo_menurun.empty:
    aset_id = df_saldo_menurun['aset_id']
    umur_ekonomis = df_saldo_menurun['umur_ekonomis']
    depresiasi_tahun_pertama =
        df_saldo_menurun['depresiasi_tahun_pertama']

    # Buat visualisasi 3D Bar Chart
    try:
        fig = plt.figure(figsize=(12, 8))
        ax = fig.add_subplot(111, projection='3d')

        # Buat posisi untuk batang
        xpos = np.arange(len(aset_id))
        ypos = umur_ekonomis.values
        zpos = np.zeros_like(xpos)

        # Lebar dan kedalaman batang
        dx = 0.8 * np.ones_like(zpos)
        dy = 1.0 * np.ones_like(zpos)
        dz = depresiasi_tahun_pertama.values / 1e6 # Tinggi batang
(depresiasi dalam Juta Rupiah)

        # Warna batang (bisa disesuaikan, misalnya berdasarkan
        kategori atau nilai depresiasi)
        colors = plt.cm.plasma(depresiasi_tahun_pertama /
depresiasi_tahun_pertama.max())

        # Buat 3D bar chart
        ax.bar3d(xpos, ypos, zpos, dx, dy, dz, color=colors,
alpha=0.8)

        # Atur label sumbu
        ax.set_xlabel('Aset ID', fontsize=12)
        ax.set_ylabel('Umur Ekonomis (Tahun)', fontsize=12)
        ax.set_zlabel('Depresiasi Tahun Pertama (Juta Rupiah)',
fontsize=12)

        # Atur tick dan label sumbu X (Aset ID)
        ax.set_xticks(xpos + dx[0]/2)
        ax.set_xticklabels(aset_id)

        # Judul plot
        ax.set_title('Simulasi Depresiasi Metode Saldo Menurun (3D
Bar Chart)', fontsize=16)

        # Tambahkan colorbar jika menggunakan colormap

```

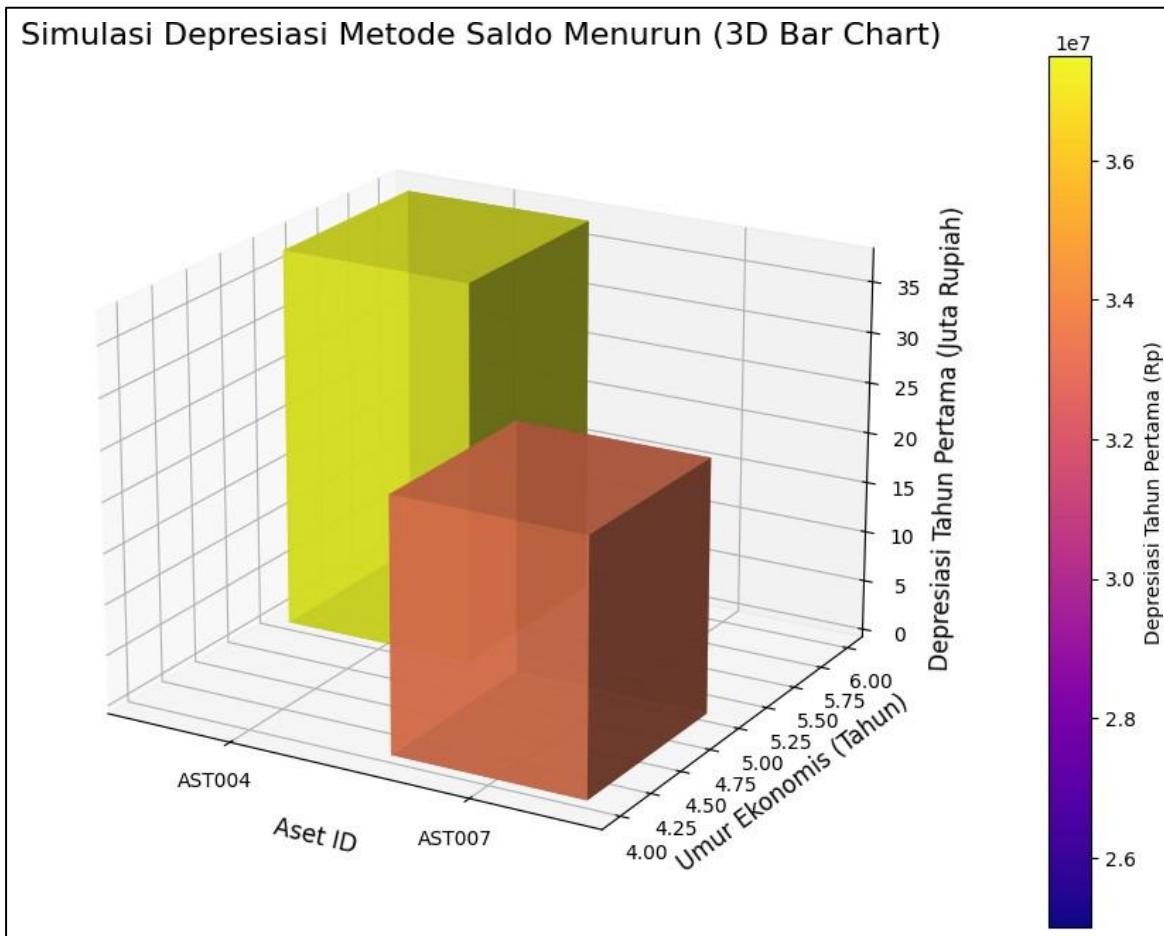
```

        cbar = fig.colorbar(plt.cm.ScalarMappable(cmap='plasma',
norm=plt.Normalize(vmin=depresiasi_tahun_pertama.min(),
vmax=depresiasi_tahun_pertama.max()), ax=ax, pad=0.1)
cbar.set_label('Depresiasi Tahun Pertama (Rp)')

# Sesuaikan sudut pandang
ax.view_init(elev=20, azim=-60) # Mengubah sudut pandang

plt.show()
except Exception as e:
    print(f"Error saat membuat visualisasi: {str(e)}")
    print("Pastikan matplotlib dan mpl_toolkits.mplot3d terinstal dengan benar.")
else:
    print("DataFrame untuk metode Saldo Menurun kosong atau tidak tersedia untuk visualisasi.")

```



aset_id	kategori	nilai_perolehan	umur_ekonomis	depresiasi_tahun_pertama	depresiasi_tahun_pertama_alternatif
AST004	Peralatan	15000000	5	37500000.0	60000000.0
AST007	Peralatan	10000000	4	25000000.0	50000000.0

Analisis:

1. Analisis BigQuery – Metode Depresiasi Saldo Menurun

Data yang ditampilkan:

- Aset ID: AST004 dan AST007
- Nilai Perolehan:
 - AST004: Rp150.000.000
 - AST007: Rp100.000.000
- Umur Ekonomis:
 - AST004: 5 tahun
 - AST007: 4 tahun

Visualisasi:

- Grafik batang menunjukkan hubungan antara umur ekonomis dan nilai perolehan.
- Warna batang (ungu muda) digunakan untuk membedakan nilai perolehan antar aset.

Keterbatasan:

- Informasi depresiasi tahunan belum muncul dalam grafik maupun tabel.
- Tidak ditampilkan kategori aset dan nilai depresiasi tahun pertama, yang padahal penting untuk analisis metode saldo menurun.
- Perlu dilakukan penambahan data di bagian matriks (Pivot Table) untuk menampilkan nilai depresiasi atau informasi akuntansi lainnya.

Kesimpulan sementara:

- BigQuery lebih bersifat sebagai alat eksplorasi awal data, namun belum cukup untuk menampilkan hasil depresiasi secara langsung tanpa konfigurasi tambahan.

2. Analisis Google Colab – Metode Depresiasi Saldo Menurun

Data yang ditampilkan:

- Aset ID: AST004 dan AST007
- Kategori Aset: Keduanya termasuk kategori *Peralatan*
- Nilai Perolehan:
 - AST004: Rp150.000.000
 - AST007: Rp100.000.000
- Umur Ekonomis:
 - AST004: 5 tahun
 - AST007: 4 tahun
- Depresiasi Tahun Pertama:
 - AST004: Rp37.500.000
 - AST007: Rp25.000.000
- Depresiasi Tahun Pertama (Alternatif):
 - AST004: Rp60.000.000
 - AST007: Rp50.000.000
(menggambarkan kemungkinan perhitungan dengan metode saldo menurun ganda atau variasi lainnya)

Visualisasi:

- 3D Bar Chart yang menunjukkan:
 - Sumbu X: Aset ID
 - Sumbu Y: Umur Ekonomis
 - Sumbu Z: Depresiasi Tahun Pertama
- Disertai color bar untuk menunjukkan intensitas nilai depresiasi.

Kelebihan:

- Semua informasi penting langsung ditampilkan dalam satu tabel dan grafik.

- Tidak memerlukan konfigurasi tambahan, sehingga efisien untuk keperluan analisis akuntansi atau pelaporan.
- Dapat menunjukkan perbedaan perhitungan depresiasi alternatif secara langsung.

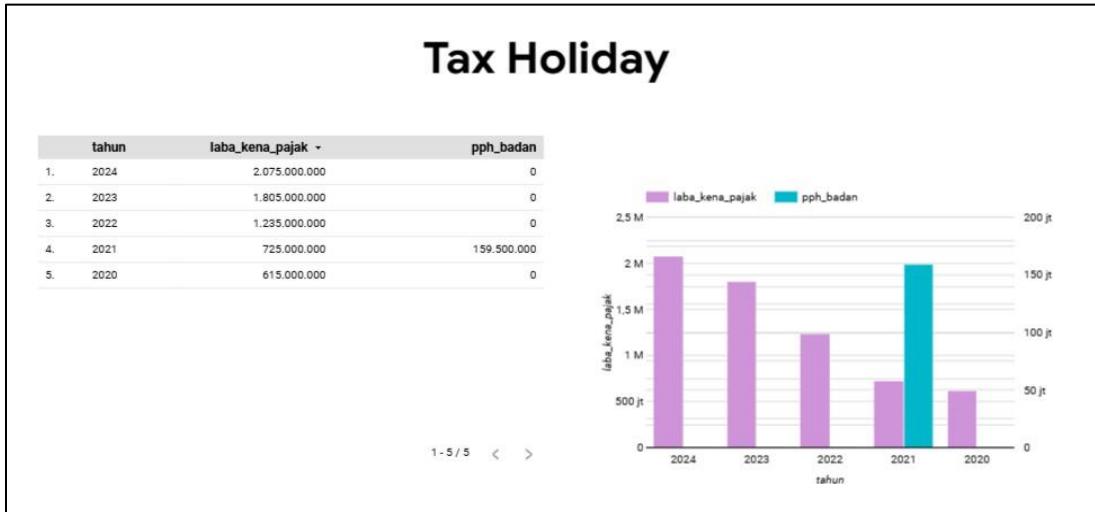
Kesimpulan

- BigQuery cocok digunakan untuk tahap awal, terutama untuk mengambil dan menampilkan data aset secara umum. Namun, informasi penting seperti depresiasi tahun pertama tidak tersedia secara otomatis dan harus ditambahkan secara manual melalui fitur matriks atau kalkulasi tambahan.
- Google Colab jauh lebih efisien untuk analisis depresiasi metode saldo menurun karena:
 - Menyediakan informasi lengkap dan langsung ditampilkan.
 - Sudah melakukan perhitungan depresiasi tahun pertama.
 - Memiliki grafik 3D yang informatif untuk menunjukkan tren dan nilai depresiasi secara visual.

Skenario Tax Holiday

1. BigQuery

```
-- C. Simulasi Tax Holiday
-- Menghitung PPh Badan berdasarkan laba kena pajak dan periode tax holiday
-- Laba kena pajak dihitung dari transaksi_keuangan, digabungkan dengan
kebijakan_fiskal
WITH LabaKotor AS (
  SELECT
    tahun,
    skenario,
    SUM(pendapatan) - SUM(beban_operasional + penyusutan) AS laba_kena_pajak
  FROM `utilitarian-web-459016-v4.PPh_Badan.transaksi_keuangan`
  GROUP BY tahun, skenario
)
SELECT
  l.tahun,
  l.skenario,
  l.laba_kena_pajak,
  CASE
    WHEN l.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir THEN 0
    ELSE l.laba_kena_pajak * k.tax_rate
  END AS pph_badan
FROM LabaKotor l
JOIN `utilitarian-web-459016-v4.PPh_Badan.kebijakan_fiskal` k
  ON l.tahun = k.tahun
ORDER BY l.tahun, l.skenario;
```



2. Google Colab

```
# Visualisasi 3D Bar Chart untuk simulasi Tax Holiday
fig = plt.figure(figsize=(14, 10))
ax = fig.add_subplot(111, projection='3d')

# Dapatkan daftar tahun dan skenario unik dari data tax holiday
years_th = df['tahun'].unique()
scenarios_th = df['skenario'].unique()

# Buat posisi untuk batang pada sumbu x (tahun) dan y (skenario)
xpos_th, ypos_th = np.meshgrid(np.arange(len(years_th)),
np.arange(len(scenarios_th)))
xpos_th = xpos_th.flatten()
ypos_th = ypos_th.flatten()
zpos_th = np.zeros(len(xpos_th))

# Lebar dan kedalaman batang
dx_th = 0.5 * np.ones_like(zpos_th)
dy_th = 0.5 * np.ones_like(zpos_th)

# Tinggi batang (PPh Badan) - gunakan data dari DataFrame df
dz_th = df['pph_badan'].values

# Mapping warna untuk setiap skenario (gunakan warna yang sudah ada atau definisikan lagi)
colors_th = {'Optimistis': 'dodgerblue', 'Moderat': 'limegreen',
'Pesimistis': 'salmon'}
scenario_colors_th = [colors_th[skenario] for skenario in df['skenario']]

# Buat 3D bar chart
```

```

ax.bar3d(xpos_th,      ypos_th,      zpos_th,      dx_th,      dy_th,      dz_th,
color=scenario_colors_th, alpha=0.8)

# Atur label sumbu
ax.set_xlabel('Tahun', fontsize=12)
ax.set_ylabel('Skenario', fontsize=12)
ax.set_zlabel('PPh Badan (Rp)', fontsize=12)

# Atur tick dan label sumbu. Ticks harus berada di tengah kelompok
# batang untuk setiap tahun dan skenario.
x_tick_positions_th = np.arange(len(years_th)) + dx_th[0]/2
ax.set_xticks(x_tick_positions_th)
ax.set_xticklabels(years_th)

y_tick_positions_th = np.arange(len(scenarios_th)) + dy_th[0]/2
ax.set_yticks(y_tick_positions_th)
ax.set_yticklabels(scenarios_th)

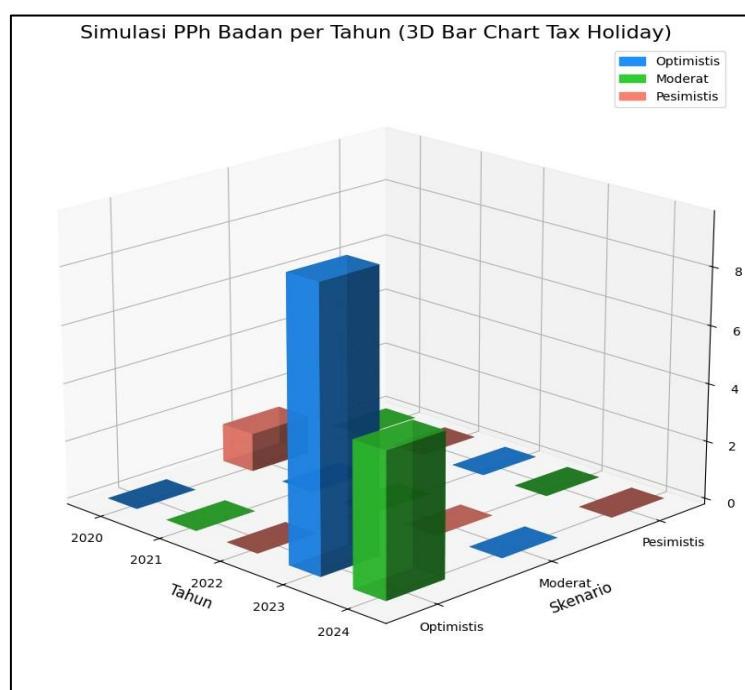
# Judul plot
ax.set_title('Simulasi PPh Badan per Tahun (3D Bar Chart Tax Holiday)', fontsize=16)

# Tambahkan legenda manual
import matplotlib.patches as mpatches
legend_patches_th = [mpatches.Patch(color=colors_th[skenario], label=skenario) for skenario in scenarios_th]
ax.legend(handles=legend_patches_th)

# Sesuaikan sudut pandang
ax.view_init(elev=20, azim=-45)

plt.show()

```



tahun	skenario	laba_kena_pajak	pph_badan
2020	Optimistis	350000000	0
2020	Moderat	205000000	0
2020	Pesimistik	60000000	0
2021	Optimistis	440000000	96800000
2021	Moderat	225000000	49500000
2021	Pesimistik	60000000	13200000
2022	Optimistis	630000000	0
2022	Moderat	435000000	0
2022	Pesimistik	190000000	0
2023	Optimistis	820000000	0
2023	Moderat	605000000	0
2023	Pesimistik	380000000	0
2024	Optimistis	910000000	0
2024	Moderat	695000000	0
2024	Pesimistik	470000000	0

Analisis:

1. Penjelasan Hasil BigQuery

Data dari BigQuery menunjukkan laporan laba kena pajak dan PPh Badan (pajak penghasilan badan) selama 5 tahun terakhir, dari tahun 2020 hingga 2024.

Penjelasan:

- Terdapat kebijakan Tax Holiday, yaitu pembebasan pajak, yang berlaku selama beberapa tahun.
- Hanya tahun 2021 yang tidak mendapat Tax Holiday, sehingga PPh Badan dikenakan sebesar Rp 159.500.000.
- Di luar tahun 2021 (yaitu 2020, 2022, 2023, dan 2024), nilai PPh Badan = 0, karena mendapatkan fasilitas pembebasan pajak.
- Laba kena pajak mengalami tren kenaikan signifikan, dari Rp 615 juta di 2020 hingga Rp 2,075 miliar di 2024.

Visualisasi:

- Bar chart memperlihatkan tingginya laba kena pajak di tiap tahun.
- Warna ungu menunjukkan laba, dan biru menunjukkan PPh Badan.
- Bar biru hanya muncul di tahun 2021, mencerminkan bahwa hanya tahun itu yang terkena pajak.

2. Penjelasan Hasil Google Colab

Data Google Colab dibuat dalam bentuk simulasi menggunakan tiga skenario: Optimistis, Moderat, dan Pesimistik. Tujuannya adalah untuk melihat dampak perbedaan asumsi terhadap pajak perusahaan.

Penjelasan:

- Untuk setiap skenario, nilai laba kena pajak bervariasi, mencerminkan kondisi ekonomi atau performa perusahaan.
- Seperti data BigQuery, pajak hanya dikenakan di tahun 2021. Tahun-tahun lain tetap 0 karena masih dalam masa Tax Holiday.
- Nilai pajak di tahun 2021 berbeda tergantung skenario:
 - Optimistis: Rp 96.800.000
 - Moderat: Rp 49.500.000
 - Pesimistik: Rp 13.200.000

Visualisasi:

- Diagram 3D bar chart menunjukkan hubungan antara tahun, skenario, dan besarnya PPh Badan.
- Warna biru (optimistik), hijau (moderat), dan merah (pesimistik) membantu membedakan masing-masing skenario.

- Grafik menunjukkan bahwa skenario optimistis menghasilkan pajak tertinggi karena laba tertinggi.

Kesimpulan:

- BigQuery cocok digunakan untuk pelaporan keuangan dan kepatuhan pajak karena data aktualnya yang sederhana dan jelas.
- Google Colab lebih cocok untuk pengambilan keputusan strategis karena mampu menyajikan berbagai skenario masa depan.
- Kedua pendekatan menunjukkan bahwa Tax Holiday sangat efektif dalam mengurangi beban pajak dan mendukung pertumbuhan laba.

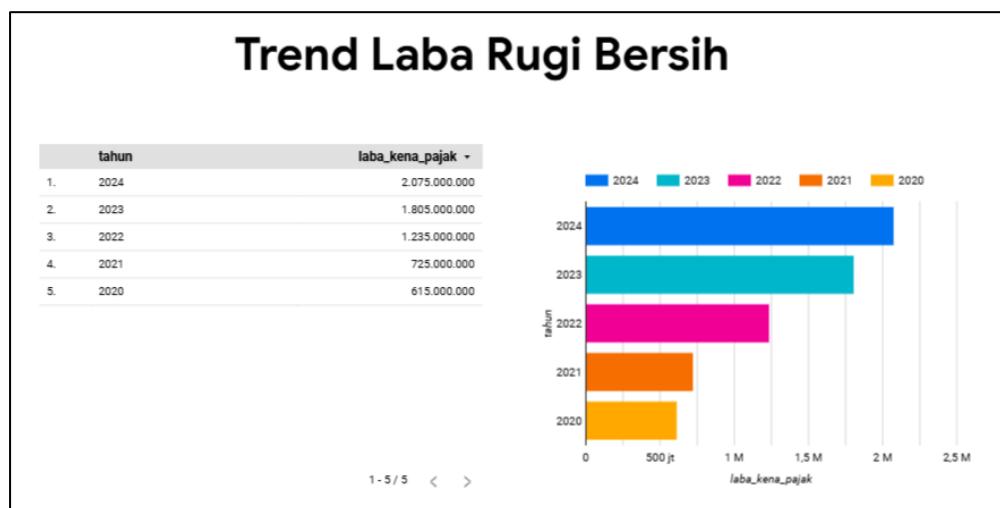
Rekomendasi:

Gunakan keduanya secara bersamaan untuk hasil terbaik. Gunakan BigQuery untuk laporan aktual ke regulator dan internal, lalu gunakan Google Colab untuk simulasi perencanaan bisnis yang mempertimbangkan berbagai kemungkinan ke depan.

Trend Laba Rugi Bersih

1. BigQuery

```
-- Menghitung laba kotor bersih untuk semua skenario
SELECT
    tahun,
    skenario,
    SUM(pendapatan) - SUM(beban_operasional + penyusutan) AS laba_kena_pajak
FROM `utilitarian-web-459016-v4.PPh_Badan.transaksi_keuangan`
GROUP BY tahun, skenario
ORDER BY tahun, skenario;
```



2. Google Colab

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np # Import numpy
```

```

# Assuming 'df' from cell MCdS1dQ0kOFX contains the 'tahun',
'skenario', and 'laba_kotor' data

# Group by year and scenario to get the laba kotor for each bar
df_grouped = df.groupby(['tahun',
'skenario'])['laba_kotor'].sum().reset_index()

# Set up the plot
fig, ax = plt.subplots(figsize=(12, 7))

# Define colors for each scenario
colors = {'Optimistis': 'skyblue', 'Moderat': 'lightcoral',
'Pesimistis': 'lightgreen'}

# Create a bar for each year and scenario
bar_width = 0.25
x_positions = np.arange(len(df_grouped['tahun'].unique()))

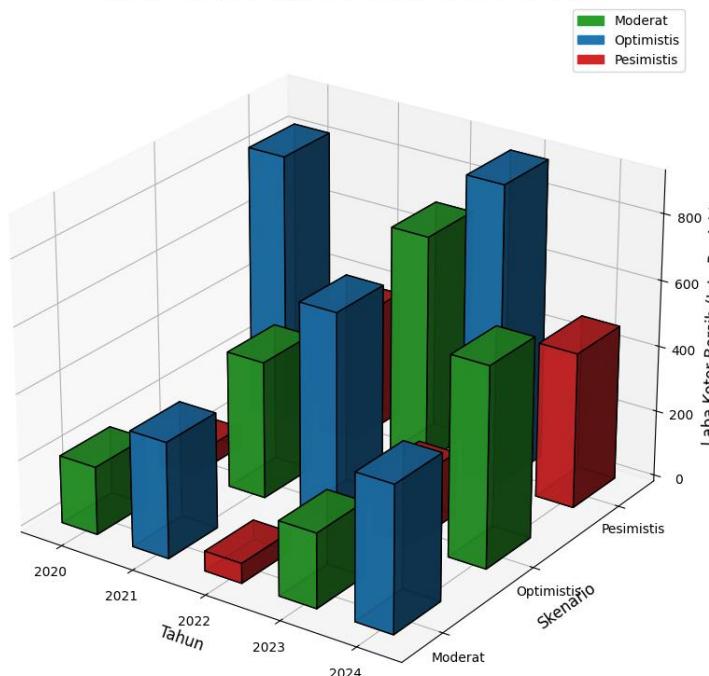
for i, skenario in enumerate(df_grouped['skenario'].unique()):
    df_skenario = df_grouped[df_grouped['skenario'] == skenario]
    # Adjust the x position for each scenario's bars within a year
    group
    x_offset = (i - len(df_grouped['skenario'].unique()) / 2 + 0.5)
    * bar_width
    ax.bar(x_positions + x_offset, df_skenario['laba_kotor'] / 1e6,
bar_width, label=skenario, color=colors[skenario])

# Set labels and title
ax.set_xlabel('Tahun')
ax.set_ylabel('Laba Kotor Bersih (Juta Rupiah)')
ax.set_title('Tren Laba Kotor Bersih per Tahun dan Skenario')
ax.set_xticks(x_positions)
ax.set_xticklabels(df_grouped['tahun'].unique())
ax.legend()
ax.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()

```

Tren Laba Kotor Bersih per Tahun (3D Bar Chart)



tahun	skenario	laba_kotor
2020	Optimistis	350000000
2020	Moderat	205000000
2020	Pessimistis	60000000
2021	Optimistis	440000000
2021	Moderat	225000000
2021	Pessimistis	60000000
2022	Optimistis	630000000
2022	Moderat	435000000
2022	Pessimistis	190000000
2023	Optimistis	820000000
2023	Moderat	605000000
2023	Pessimistis	380000000
2024	Optimistis	910000000
2024	Moderat	695000000
2024	Pessimistis	470000000

Analisis:

1. Tren Laba Kotor Bersih BigQuery

Data dari BigQuery menunjukkan tren kenaikan laba kena pajak dari tahun 2020 hingga 2024. Setiap tahun mengalami pertumbuhan yang konsisten. Pada tahun 2020, laba masih berada di angka sekitar 615 juta rupiah. Angka ini meningkat perlakuan di tahun 2021, kemudian melonjak cukup signifikan pada tahun 2022 dan terus naik tajam hingga mencapai lebih dari 2 miliar rupiah di tahun 2024. Kenaikan ini menunjukkan kinerja keuangan yang semakin membaik dari waktu ke waktu, dan mencerminkan pertumbuhan usaha yang stabil dan positif. Data ini bersifat konkret, didasarkan pada realisasi yang aktual, sehingga sangat cocok digunakan untuk pelaporan keuangan resmi, audit, atau evaluasi kinerja tahunan.

2. Tren Laba Kotor Bersih Google Colab

Sementara itu, data dari Google Colab menyajikan pendekatan yang berbeda. Di sini digunakan tiga skenario dalam memproyeksikan laba kotor bersih, yaitu skenario optimistis, moderat, dan pesimistik. Ketiga skenario ini menggambarkan potensi kondisi keuangan berdasarkan asumsi yang berbeda-beda. Skenario optimistis menunjukkan tren kenaikan laba yang paling tinggi dan stabil sepanjang lima tahun. Skenario moderat juga memperlihatkan pertumbuhan, meskipun tidak sebesar

skenario optimistis. Sedangkan skenario pesimistik menampilkan tren yang lebih konservatif, namun tetap mengalami peningkatan secara bertahap dari tahun ke tahun. Pendekatan ini sangat berguna untuk keperluan perencanaan dan pengambilan keputusan, karena dapat memperkirakan potensi keuntungan maupun risiko yang mungkin dihadapi berdasarkan berbagai kondisi pasar.

Perbandingan

Perbedaan utama antara kedua pendekatan ini terletak pada jenis data dan tujuan penggunaannya. Data dari BigQuery bersifat historis dan aktual, sehingga sangat cocok untuk pelaporan dan analisis keuangan yang sudah terjadi. Sebaliknya, data dari Google Colab bersifat proyektif dengan mempertimbangkan berbagai kemungkinan, yang lebih tepat digunakan untuk merancang strategi bisnis dan merencanakan anggaran ke depan.

Berdasarkan kebutuhan, jika tujuan utama adalah menyampaikan laporan realisasi keuangan yang valid dan bisa dipertanggungjawabkan, maka data dari BigQuery lebih tepat digunakan. Namun, jika yang dibutuhkan adalah alat bantu untuk merancang strategi ke depan dan mengantisipasi risiko berdasarkan berbagai kemungkinan, maka data dari Google Colab jauh lebih bermanfaat.

Kesimpulan

Pendekatan terbaik adalah menggunakan keduanya secara bersamaan: BigQuery sebagai dasar evaluasi atas kinerja aktual, dan Google Colab sebagai alat perencanaan strategis ke depan. Kombinasi ini akan memberikan gambaran yang menyeluruh dan seimbang dalam pengambilan keputusan keuangan.

Perbandingan PPh Badan

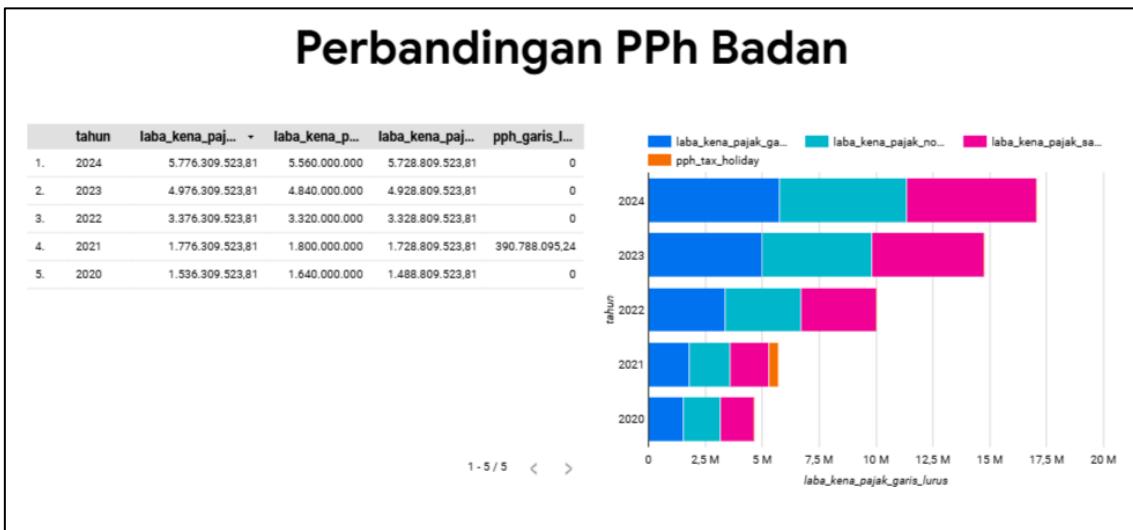
1. BigQuery

```
-- Perbandingan PPh Badan: Normal (Moderat), Tax Holiday, Metode Depresiasi
WITH LabaKotor AS (
    SELECT
        t.tahun,
        t.skenario,
        SUM(t.pendapatan) - SUM(t.beban_operasional) + t.penyusutan) AS
laba_kena_pajak_normal,
        SUM(t.pendapatan) - SUM(t.beban_operasional) - SUM(
            CASE
                WHEN a.metode = 'Garis_Lurus' THEN a.nilai_perolehan / a.umur_ekonomis
                WHEN a.metode = 'Saldo_Menurun' THEN a.nilai_perolehan * 0.25
                ELSE 0
            END
        ) AS laba_kena_pajak_garis_lurus,
        SUM(t.pendapatan) - SUM(t.beban_operasional) - SUM(
            CASE
                WHEN a.metode = 'Garis_Lurus' THEN a.nilai_perolehan / a.umur_ekonomis
                WHEN a.metode = 'Saldo_Menurun' THEN a.nilai_perolehan * (2.0 /
a.umur_ekonomis)
                ELSE 0
            END
        ) AS laba_kena_pajak saldo_menurun
    FROM `utilitarian-web-459016-v4.PPh_Badan.transaksi_keuangan` t
    CROSS JOIN `utilitarian-web-459016-v4.PPh_Badan.aset_tetap` a
```

```

        WHERE t.skenario = 'Moderat'
        GROUP BY t.tahun, t.skenario
    ),
PPh AS (
    SELECT
        l.tahun,
        l.skenario,
        l.laba_kena_pajak_normal,
        l.laba_kena_pajak_garis_lurus,
        l.laba_kena_pajak_saldo_menurun,
        l.laba_kena_pajak_normal * k.tax_rate AS pph_normal,
        CASE
            WHEN l.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir THEN 0
            ELSE l.laba_kena_pajak_normal * k.tax_rate
        END AS pph_tax_holiday,
        CASE
            WHEN l.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir THEN 0
            ELSE l.laba_kena_pajak_garis_lurus * k.tax_rate
        END AS pph_garis_lurus,
        CASE
            WHEN l.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir THEN 0
            ELSE l.laba_kena_pajak_saldo_menurun * k.tax_rate
        END AS pph_saldo_menurun
    FROM LabaKotor l
    JOIN `utilitarian-web-459016-v4.PPh_Badan.kebijakan_fiskal` k
        ON l.tahun = k.tahun
)
SELECT
    tahun,
    skenario,
    laba_kena_pajak_normal,
    laba_kena_pajak_garis_lurus,
    laba_kena_pajak_saldo_menurun,
    pph_normal,
    pph_tax_holiday,
    pph_garis_lurus,
    pph_saldo_menurun
FROM PPh
ORDER BY tahun;

```



2. Google Colab

```
try:
    from google.cloud import bigquery
    import pandas as pd
    import matplotlib.pyplot as plt
    from mpl_toolkits.mplot3d import Axes3D
    from google.colab import files
    import os
except ImportError as e:
    print(f"Error: Library tidak terinstal. Silakan jalankan: !pip
install {str(e).split()[-1]}")
    !pip install google-cloud-bigquery pandas matplotlib
    from google.cloud import bigquery
    import pandas as pd
    import matplotlib.pyplot as plt
    from mpl_toolkits.mplot3d import Axes3D
    from google.colab import files
    import os

def upload_credentials():
    print("Unggah file kredensial JSON Anda:")
    uploaded = files.upload()
    if not uploaded:
        raise ValueError("File kredensial tidak diunggah!")
    credential_file = list(uploaded.keys())[0]
    os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = credential_file
    return credential_file

try:
    client = bigquery.Client()
except Exception as e:
    print(f"Error saat inisialisasi BigQuery: {str(e)}")
    print("Mari kita coba unggah kredensial JSON.")
    upload_credentials()
    client = bigquery.Client()

query = """
WITH LabaKotor AS (
    SELECT
        t.tahun,
        t.skenario,
        SUM(t.pendapatan) - SUM(t.beban_operasional + t.penyusutan) AS
laba_kena_pajak_normal,
        SUM(t.pendapatan) - SUM(t.beban_operasional) - SUM(
            CASE
                WHEN a.metode = 'Garis_Lurus' THEN a.nilai_perolehan /
a.umur_ekonomis
                WHEN a.metode = 'Saldo_Menurun' THEN a.nilai_perolehan * 0.25
                ELSE 0
            
```

```

        END
    ) AS laba_kena_pajak_garis_lurus,
    SUM(t.pendapatan) - SUM(t.beban_operasional) - SUM(
        CASE
            WHEN a.metode = 'Garis_Lurus' THEN a.nilai_perolehan /
a.umur_ekonomis
            WHEN a.metode = 'Saldo_Menurun' THEN a.nilai_perolehan * (2.0
/ a.umur_ekonomis)
            ELSE 0
        END
    ) AS laba_kena_pajak saldo_menurun
FROM `utilitarian-web-459016-v4.PPh_Badan.transaksi_keuangan` t
CROSS JOIN `utilitarian-web-459016-v4.PPh_Badan.aset_tetap` a
WHERE t.skenario = 'Moderat'
GROUP BY t.tahun, t.skenario
),
PPh AS (
    SELECT
        l.tahun,
        l.skenario,
        l.laba_kena_pajak_normal,
        l.laba_kena_pajak_garis_lurus,
        l.laba_kena_pajak saldo_menurun,
        l.laba_kena_pajak_normal * k.tax_rate AS pph_normal,
        CASE
            WHEN l.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir
THEN 0
            ELSE l.laba_kena_pajak_normal * k.tax_rate
        END AS pph_tax_holiday,
        CASE
            WHEN l.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir
THEN 0
            ELSE l.laba_kena_pajak_garis_lurus * k.tax_rate
        END AS pph_garis_lurus,
        CASE
            WHEN l.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir
THEN 0
            ELSE l.laba_kena_pajak saldo_menurun * k.tax_rate
        END AS pph saldo_menurun
    FROM LabaKotor l
    JOIN `utilitarian-web-459016-v4.PPh_Badan.kebijakan_fiskal` k
        ON l.tahun = k.tahun
)
SELECT
    tahun,
    skenario,
    laba_kena_pajak_normal,
    laba_kena_pajak_garis_lurus,

```

```

laba_kena_pajak_saldo_menurun,
pph_normal,
pph_tax_holiday,
pph_garis_lurus,
pph_saldo_menurun
FROM PPh
ORDER BY tahun;
"""

try:
    query_job = client.query(query)
    df = query_job.to_dataframe()
except Exception as e:
    print(f"Error saat menjalankan query: {str(e)}")
    print("Menggunakan data cadangan...")
    # Data cadangan dihitung berdasarkan CSV
    data = {
        'tahun': [2020, 2021, 2022, 2023, 2024],
        'skenario': ['Moderat', 'Moderat', 'Moderat', 'Moderat', 'Moderat'],
        'laba_kena_pajak_normal': [205000000, 225000000, 435000000, 605000000, 695000000],
        'laba_kena_pajak_garis_lurus': [205000000 - 32083333.33, 225000000 - 32083333.33, 435000000 - 32083333.33, 605000000 - 32083333.33, 695000000 - 32083333.33],
        'laba_kena_pajak_saldo_menurun': [205000000 - 46250000, 225000000 - 46250000, 435000000 - 46250000, 605000000 - 46250000, 695000000 - 46250000],
        'pph_normal': [0, 49500000, 0, 0, 0],
        'pph_tax_holiday': [0, 0, 0, 0, 0],
        'pph_garis_lurus': [0, 0, 0, 0, 0],
        'pph_saldo_menurun': [0, 0, 0, 0, 0]
    }
    df = pd.DataFrame(data)

print("Data Perbandingan PPh:")
print(df)

try:
    fig = plt.figure(figsize=(12, 8))
    ax = fig.add_subplot(111, projection='3d')

    # Plot untuk setiap jenis PPh
    methods = ['pph_normal', 'pph_tax_holiday', 'pph_garis_lurus',
    'pph_saldo_menurun']
    colors = ['b', 'g', 'r', 'purple']
    labels = ['Normal', 'Tax Holiday', 'Garis Lurus', 'Saldo Menurun']

```

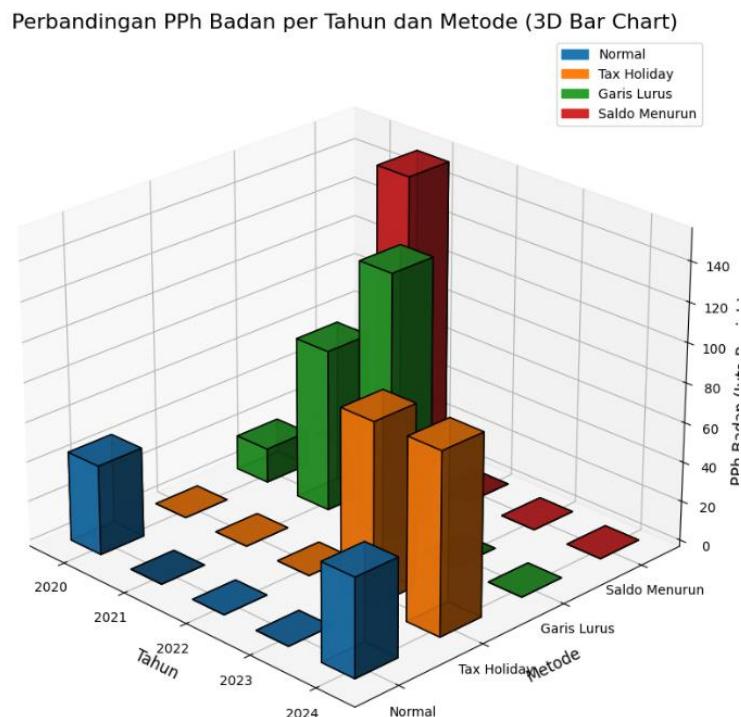
```

for i, method in enumerate(methods):
    ax.scatter(
        df['tahun'],
        df[method],
        [i+1] * len(df),
        c=colors[i],
        marker='o',
        s=100,
        label=labels[i]
    )
    ax.plot(
        df['tahun'],
        df[method],
        [i+1] * len(df),
        c=colors[i],
        linestyle='--'
    )

ax.set_xlabel('Tahun')
ax.set_ylabel('PPh Badan (Rp)')
ax.set_zlabel('Metode (1=Normal, 2=Tax Holiday, 3=Garis Lurus,  
4=Saldo Menurun)')
ax.set_zticks([1, 2, 3, 4])
ax.set_zticklabels(labels)
ax.set_title('Perbandingan PPh Badan (3D)')
ax.legend()
plt.show()

except Exception as e:
    print(f"Error saat membuat visualisasi: {str(e)}")

```



tahun	skenario	pph_normal	pph_tax_holiday	pph_garis_lurus	pph_saldo_menurun
2020	Moderat	0	0	0	0
2021	Moderat	49500000	0	0	0
2022	Moderat	0	0	0	0
2023	Moderat	0	0	0	0
2024	Moderat	0	0	0	0

Analisis:

1. Analisis Data dari BigQuery

- Data mencakup tahun 2020 hingga 2024 dengan tren laba kena pajak yang terus meningkat setiap tahun.
- Laba kena pajak pada tahun 2020 sebesar sekitar Rp 1,6 miliar dan meningkat hingga lebih dari Rp 5,7 miliar pada tahun 2024.
- Kenaikan laba ini menunjukkan performa keuangan perusahaan yang tumbuh secara stabil.
- Beberapa metode perhitungan PPh Badan ditampilkan, yaitu: Normal, Tax Holiday, Garis Lurus, dan Saldo Menurun.
- Metode Garis Lurus menunjukkan tren nilai pajak yang konsisten meningkat setiap tahun, sesuai dengan kenaikan laba.
- Tahun 2021 tercatat sebagai satu-satunya tahun yang menggunakan Tax Holiday, yang menghasilkan pengurangan pajak hingga Rp 390 juta.
- Metode Saldo Menurun menunjukkan nilai PPh yang cenderung kecil, cocok untuk percepatan biaya di awal masa pakai aset.
- Visualisasi grafik bar memperlihatkan distribusi yang jelas dan proporsional antar metode dari tahun ke tahun.

2. Analisis Data dari Google Colab

- Data bersumber dari simulasi menggunakan skenario “Moderat” untuk lima tahun, dari 2020 hingga 2024.
- Empat metode perhitungan juga digunakan, yaitu Normal, Tax Holiday, Garis Lurus, dan Saldo Menurun.
- Hanya metode Normal di tahun 2021 yang menghasilkan nilai PPh sebesar Rp 495 juta, sementara lainnya bernilai nol.
- Nilai PPh nol menunjukkan kemungkinan besar karena adanya pengaruh simulasi seperti depresiasi penuh, tax shield, atau belum adanya laba kena pajak.
- Google Colab digunakan untuk skenario dan proyeksi, bukan data aktual, sehingga lebih cocok untuk perencanaan strategis.
- Visualisasi 3D memperlihatkan perbandingan antar metode dan tahun, namun secara data numerik kurang detail dibandingkan BigQuery.

Kesimpulan dan Rekomendasi

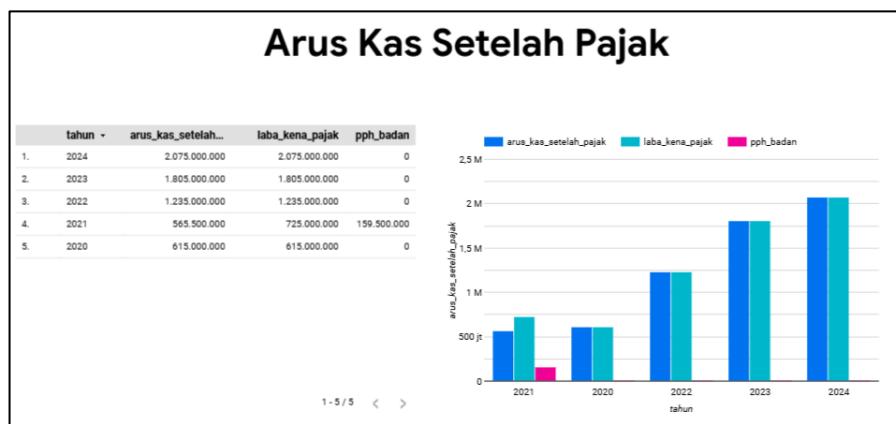
- Metode BigQuery lebih baik untuk keperluan pelaporan, perhitungan PPh aktual, dan analisis keuangan berbasis data nyata.

- Metode Google Colab lebih sesuai untuk perencanaan ke depan, pengujian asumsi, dan melihat potensi dampak kebijakan fiskal.
- Jika hanya memilih satu, BigQuery adalah yang paling direkomendasikan karena hasilnya akurat, lengkap, dan dapat langsung digunakan untuk kewajiban perpajakan.
- Namun, kombinasi keduanya akan sangat optimal: gunakan BigQuery untuk perhitungan nyata dan Colab untuk simulasi keputusan strategis jangka panjang.

Arus Kas Setelah Pajak Per Tahun dan Skenario

1. BigQuery

```
-- Analisis arus kas setelah pajak
WITH LabaKotor AS (
  SELECT
    tahun,
    skenario,
    SUM(pendapatan) - SUM(beban_operasional + penyusutan) AS laba_kena_pajak
  FROM `utilitarian-web-459016-v4.PPh_Badan.transaksi_keuangan`
  GROUP BY tahun, skenario
)
SELECT
  l.tahun,
  l.skenario,
  l.laba_kena_pajak,
  CASE
    WHEN l.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir THEN 0
    ELSE l.laba_kena_pajak * k.tax_rate
  END AS pph_badan,
  l.laba_kena_pajak - CASE
    WHEN l.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir THEN 0
    ELSE l.laba_kena_pajak * k.tax_rate
  END AS arus_kas_setelah_pajak
FROM LabaKotor l
JOIN `utilitarian-web-459016-v4.PPh_Badan.kebijakan_fiskal` k
  ON l.tahun = k.tahun
ORDER BY l.tahun, l.skenario;
```



2. Google Colab

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

```

import numpy as np
import pandas as pd

# Assuming 'df_final' from cell 8bc9e6ad contains the 'tahun',
'skenario', and 'arus_kas_setelah_pajak' data

if df_final is not None and not df_final.empty:
    fig = plt.figure(figsize=(14, 10))
    ax = fig.add_subplot(111, projection='3d')

    # Get the years and scenarios from the DataFrame
    years = df_final['tahun'].unique()
    scenarios = df_final['skenario'].unique()

    # Create positions for the bars on the x (year) and y (skenario)
    # axes
    xpos,     ypos      =     np.meshgrid(np.arange(len(years)),
    np.arange(len(scenarios)))
    xpos = xpos.flatten()
    ypos = ypos.flatten()
    zpos = np.zeros(len(xpos))

    # Width and depth of the bars
    dx = 0.5 * np.ones_like(xpos)
    dy = 0.5 * np.ones_like(ypos)

    # Heights of the bars (arus kas setelah pajak)
    # Need to ensure the order of dz matches the order of xpos and
    ypos
    # Group by year and scenario to get the correct order and values
    df_grouped      =      df_final.groupby(['tahun',
    'skenario'])['arus_kas_setelah_pajak'].sum().reset_index()
    dz = df_grouped['arus_kas_setelah_pajak'].values / 1e6 # Height
    of bars (in Millions)

    # Mapping colors for each scenario
    colors = {'Optimistis': '#AEC7E8', 'Moderat': '#98DF8A',
    'Pesimistis': '#FF9896'} # Example pastel colors
    # Create a list of colors corresponding to each bar based on the
    scenario in df_grouped
    scenario_colors = [colors[skenario] for skenario in
    df_grouped['skenario']]

    # Create 3D bar chart
    ax.bar3d(xpos, ypos, zpos, dx, dy, dz, color=scenario_colors,
    alpha=0.8)

    # Set axis labels

```

```

        ax.set_xlabel('Tahun', fontsize=12)
        ax.set_ylabel('Skenario', fontsize=12)
        ax.set_zlabel('Arus Kas Setelah Pajak (Juta Rupiah)', fontsize=12)

        # Set ticks and labels for axes
        x_tick_positions = np.arange(len(years)) + dx[0]/2
        ax.set_xticks(x_tick_positions)
        ax.set_xticklabels(years)

        y_tick_positions = np.arange(len(scenarios)) + dy[0]/2
        ax.set_yticks(y_tick_positions)
        ax.set_yticklabels(scenarios)

        # Title
        ax.set_title('Tren Arus Kas Setelah Pajak per Tahun (3D Bar Chart)', fontsize=16)

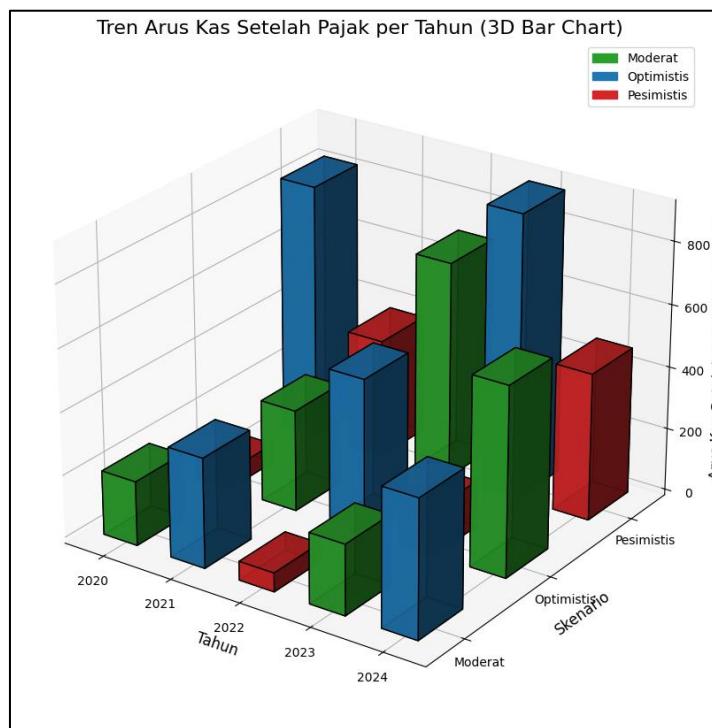
        # Add manual legend
        import matplotlib.patches as mpatches
        legend_patches = [mpatches.Patch(color=colors[skenario], label=skenario) for skenario in scenarios]
        ax.legend(handles=legend_patches)

        # Adjust view angle
        ax.view_init(elev=20, azim=-45)

        plt.show()
    else:
        print("DataFrame kosong atau tidak tersedia untuk visualisasi.")

```

tahun	skenario	laba_kena_pajak	pph_badan	arus_kas_setelah_pajak
2020	Moderat	205000000	0.0	205000000.0
2020	Optimistik	350000000	0.0	350000000.0
2020	Pessimistis	60000000	0.0	60000000.0
2021	Moderat	225000000	0.0	225000000.0
2021	Optimistik	440000000	0.0	440000000.0
2021	Pessimistis	60000000	0.0	60000000.0
2022	Moderat	415000000	91300000.0	323700000.0
2022	Optimistik	630000000	138600000.0	491400000.0
2022	Pessimistis	190000000	41800000.0	148200000.0
2023	Moderat	605000000	0.0	605000000.0
2023	Optimistik	820000000	0.0	820000000.0
2023	Pessimistis	380000000	0.0	380000000.0
2024	Moderat	695000000	0.0	695000000.0
2024	Optimistik	910000000	0.0	910000000.0
2024	Pessimistis	470000000	0.0	470000000.0



Analisis:

1. Analisis Data dari BigQuery

Struktur Data:

- Data disajikan untuk tahun 2020 sampai 2024.
- Setiap tahun memiliki tiga skenario: Moderat, Optimistis, dan Pesimistis.
- Variabel yang dianalisis:
 - laba_kena_pajak (laba perusahaan sebelum pajak)
 - pph_badan (pajak penghasilan badan)
 - arus_kas_setelah_pajak (laba setelah dikurangi pajak)

Temuan Utama:

- Pada tahun 2020 dan 2021, seluruh skenario menunjukkan bahwa tidak ada pajak penghasilan badan (PPh Badan = 0), menandakan laba yang dihasilkan mungkin masih di bawah batas kena pajak atau terdapat insentif pajak.
- Mulai tahun 2022, terdapat beban pajak untuk skenario Moderat dan Optimistis. Sementara skenario Pesimistis tetap tidak dikenakan pajak karena laba rendah.
- Arus kas setelah pajak paling tinggi konsisten berada di skenario Optimistis untuk semua tahun.
- Skenario Pesimistis menunjukkan arus kas paling rendah setiap tahunnya, meskipun tidak dikenai pajak.
- Peningkatan arus kas terlihat jelas dari tahun ke tahun, terutama di skenario Optimistis yang mencapai puncak pada 2024.

Contoh Detail Tahun 2022:

- Skenario Moderat:

Laba kena pajak = 415.000.000
PPh Badan = 91.300.000
Arus kas setelah pajak = 323.700.000
- Skenario Optimistis:

Laba kena pajak = 630.000.000
PPh Badan = 138.600.000
Arus kas setelah pajak = 491.400.000

- Skenario Pesimistis:

Laba kena pajak = 190.000.000
 PPh Badan = 41.800.000
 Arus kas setelah pajak = 148.200.000

2. Analisis Data dari Google Colab

Struktur Data:

- Menyediakan data arus kas setelah pajak dari tahun 2020 sampai 2024.
- Hanya satu skenario ditampilkan (tidak ada variasi skenario optimistis/moderat/pesimistis).
- Variabel yang ditampilkan:
 - arus_kas_setelah_pajak
 - laba_kena_pajak
 - pph_badan

Temuan Utama:

- Pada tahun 2020, 2022, 2023, dan 2024, nilai pajak penghasilan badan adalah nol. Artinya, pada tahun-tahun tersebut perusahaan tidak membayar PPh.
- Tahun 2021 menunjukkan satu-satunya tahun di mana ada PPh Badan, yaitu sebesar 159.500.000 dari laba sebesar 725.000.000. Ini menunjukkan adanya laba signifikan dan mungkin mulai melampaui ambang kena pajak.
- Arus kas setelah pajak meningkat dari 615 juta pada 2020 menjadi 2,075 miliar pada 2024. Ini menunjukkan pertumbuhan yang stabil dari waktu ke waktu.

Detail Per Tahun:

- 2020: Arus kas setelah pajak = 615.000.000
- 2021: Arus kas setelah pajak = 565.500.000 (terkena pajak)
- 2022: Arus kas setelah pajak = 1.235.000.000
- 2023: Arus kas setelah pajak = 1.805.000.000
- 2024: Arus kas setelah pajak = 2.075.000.000

Kesimpulan

- Data dari BigQuery lebih unggul karena menyediakan simulasi skenario (optimistis, moderat, pesimistis), memberikan gambaran risiko dan potensi laba secara komprehensif. Cocok untuk perencanaan keuangan jangka panjang dan analisis sensitivitas.
- Data dari Google Colab lebih sederhana, tidak menyediakan variasi skenario, dan kurang tepat jika digunakan untuk pengambilan keputusan berbasis risiko. Namun cocok untuk laporan singkat atau gambaran umum pertumbuhan arus kas.
- Jika tujuan analisis adalah pengambilan keputusan strategis, evaluasi skenario keuangan, dan analisis resiko, maka data dari BigQuery adalah pilihan terbaik.
- Jika tujuan hanya untuk menyusun laporan tahunan atau paparan ringkas, data Google Colab bisa digunakan sebagai pelengkap.

Kesimpulan

1. BigQuery – Cocok untuk Analisis Faktual dan Pelaporan

BigQuery sangat efektif digunakan untuk menyajikan data yang bersifat historis dan aktual, sehingga:

- Memudahkan dalam menghitung PPh Badan berdasarkan data keuangan yang nyata.
- Memberikan visualisasi sederhana (tabel dan diagram batang 2D) yang mudah dibaca dan cocok untuk evaluasi cepat.
- Ideal untuk keperluan audit, pelaporan pajak resmi, serta analisis kinerja masa lalu.

Contohnya, pada simulasi laba rugi dan depresiasi, BigQuery menampilkan data nyata dari transaksi keuangan dan aset tetap, yang sangat mendukung penyusunan laporan akuntansi dan perpajakan.

2. Google Colab – Kuat dalam Simulasi dan Visualisasi Interaktif

Sementara itu, Google Colab berfungsi sebagai alat analisis prediktif dan visual yang sangat berguna untuk:

- Melakukan simulasi multi-skenario (optimistis, moderat, pesimistik) terhadap laba, depresiasi, PPh Badan, dan arus kas.
- Memberikan grafik 3D yang interaktif dan menarik secara visual, sehingga sangat cocok digunakan untuk presentasi atau perencanaan strategis.
- Membantu perusahaan untuk memahami dampak kebijakan fiskal seperti tax holiday secara lebih realistik dan variatif.

Dengan menggunakan pendekatan ini, perusahaan bisa melihat perbandingan antar metode depresiasi, dampak skenario ekonomi terhadap pajak, dan tren arus kas secara visual.

3. Perbandingan dan Alasan Memilih

- Jika mengutamakan akurasi dan kepatuhan pajak, BigQuery adalah pilihan terbaik karena data yang dihasilkan bersifat konkret dan langsung bisa digunakan.
- Namun, jika tujuannya adalah mengambil keputusan jangka panjang berdasarkan kondisi yang belum pasti, maka Google Colab lebih unggul karena fleksibilitas dan kelengkapan skenario.

Pemilihan Alat yang Tepat untuk Analisis Data

- **BigQuery** merupakan pilihan yang tepat ketika bekerja dengan data berukuran besar dan membutuhkan proses analisis yang cepat menggunakan bahasa SQL. Layanan ini sangat berguna ketika semua data disimpan di cloud, sehingga tidak perlu diunduh terlebih dahulu. BigQuery juga memudahkan dalam membuat visualisasi langsung dari hasil query dan sangat cocok digunakan untuk kebutuhan analisis yang skalanya besar serta berbasis cloud.
- Di sisi lain, **Google Colab** lebih sesuai digunakan saat ingin melakukan pemrograman dengan Python, terutama untuk data berukuran kecil hingga menengah. Colab memberikan kebebasan dalam menulis kode, mengolah data, membuat grafik, atau melakukan analisis yang membutuhkan logika pemrograman yang lebih kompleks. Dengan Colab, kita dapat mengakses file dari Google Drive dan menjalankan kode secara interaktif, sehingga cocok untuk eksperimen, simulasi, atau pengolahan data yang lebih mendalam.