

**LAPORAN DATA ANALISIS PPH BADAN
MENGUNAKAN BIGQUERY DAN GOOGLE COLAB**

Disusun untuk Memenuhi Tugas Mata Kuliah Pengkodean & Pemrograman

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



Disusun Oleh:

Muhamad Daffa Sadega Hakim 12030123140332

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

A. Transaksi Keuangan

tahun	pendapatan	beban_operasional	penyusutan	skenario
2020	1200000000	800000000	100000000	baseline
2021	1350000000	850000000	100000000	baseline
2022	1500000000	900000000	100000000	baseline
2023	1600000000	950000000	100000000	baseline
2024	1700000000	1000000000	100000000	baseline
2025	1100000000	750000000	100000000	baseline
2020	1200000000	780000000	100000000	optimis
2021	1300000000	800000000	100000000	optimis
2022	1400000000	820000000	100000000	optimis
2023	1500000000	850000000	100000000	optimis
2024	1600000000	870000000	100000000	optimis
2025	1700000000	900000000	100000000	optimis

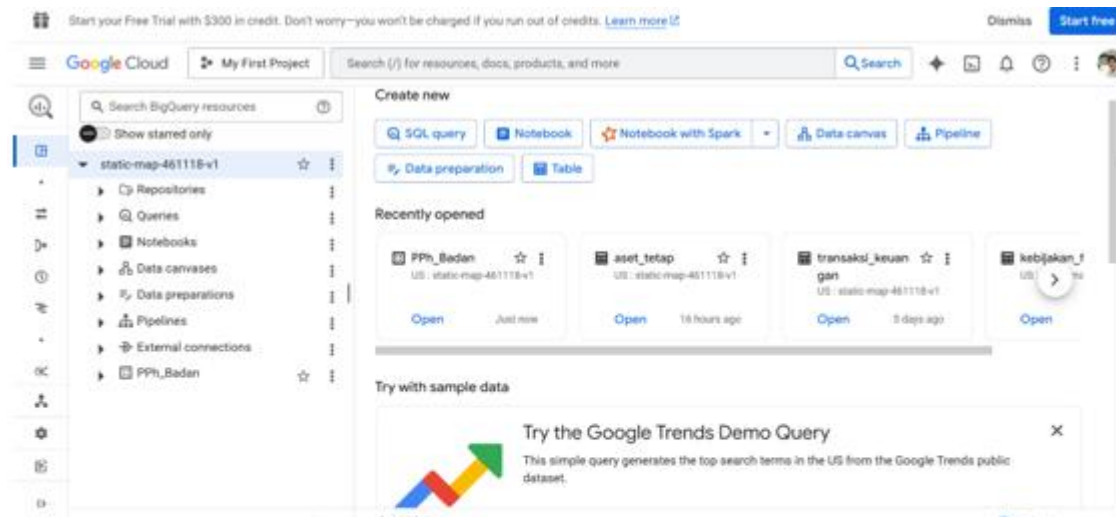
B. Aset Tetap

`	kategori	nilai_perolehan	umur_ekonomis	metode
AST001	Bangunan	5000000000	20	garis lurus
AST002	Mesin	2000000000	10	saldo menurun
AST003	Kendaraan	1000000000	5	garis lurus
AST004	Peralatan	750000000	7	saldo menurun
AST005	Perangkat	500000000	3	garis lurus

C. Kebijakan Fiskal

tahun	tax_rate	tax_holiday_awal	tax_holiday_akhir
2020	0.25	2020	2022
2021	0.24	2021	2023
2022	0.23		
2023	0.22		
2024	0.22	2023	2024
2025	0.21		

1. Mengunggah ke Big Query



Berikut dataset CSV yang saya unggah ke Query

▼	📄 PPh_Badan	☆	⋮
	📄 aset_tetap	☆	⋮
	📄 kebijakan_fiskal	☆	⋮
	📄 transaksi_keuangan	☆	⋮

2. Mmbuat Kode SQL

saya membuat kode SQL yang akan digunakan untuk melakukan simulasi perhitungan PPh Badan. Sumber data yang digunakan adalah file CSV seperti yang tertera pada tahapan sebelumnya. Setelah mendapatkan kode SQL, saya memasukkannya ke dalam BigQuery. Prompt yang berikan:

Hasil SQL yang dibuat

A. Simulasi Laba/Rugi Tiap Skenario

```
A. Simulasi Laba/Rugi Tiap Skenario
Query Asli:
sql
SELECT
    tahun,
    SUM(pendapatan) - SUM(beban_operasional + penyusutan) AS laba_kotor
FROM project.dataset.transaksi
WHERE skenario = 'normal'
GROUP BY tahun
ORDER BY tahun;
```

Job information		Results	Chart	JSON	Execution details	Execution graph
Row	tahun	laba_kotor				
1	2020	300000000				
2	2021	400000000				
3	2022	500000000				
4	2023	550000000				
5	2024	600000000				
6	2025	250000000				

B. Simulasi Depresiasi

B. Simulasi Depresiasi

1. Metode Garis Lurus

Query Asli:

sql

Collapse

Wrap

Copy

```
SELECT
  aset_id,
  nilai_perolehan,
  umur_ekonomis,
  nilai_perolehan / umur_ekonomis AS depresiasi_tahunan
FROM project.dataset.aset
WHERE metode = 'garis_lurus';
```

Job information		Results	Chart	JSON	Execution details	Execution graph
Row	aset_id	nilai_perolehan	umur_ekonomis	depresiasi_tahunan		
1	AST001	500000000	20	250000000.0		
2	AST003	1000000000	5	200000000.0		
3	AST005	500000000	3	166666666.6666...		

2. Metode Saldo Menurun

Query Asli:

sql

Collapse

Wrap

Copy

```
SELECT
  aset_id,
  nilai_perolehan,
  umur_ekonomis,
  ROUND(nilai_perolehan * 0.25, 2) AS depresiasi_tahun_pertama
FROM project.dataset.aset
WHERE metode = 'saldo_menurun';
```

Row	aset_id	nilai_perolehan	umur_ekonomis	depresiasi_tahun...	
1	AST002	2000000000	10	500000000.0	
2	AST004	750000000	7	187500000.0	

C. Simulasi PPh Badan dengan Tax Holiday (2023–2027 bebas pajak)

C. Simulasi Tax Holiday

Query Asli:

sql

✕ Collapse ⇅ Wrap 📋 Copy

```
SELECT
  tahun,
  laba_kena_pajak,
  CASE
    WHEN tahun BETWEEN 2023 AND 2027 THEN 0
    ELSE laba_kena_pajak * 0.22
  END AS pph_badan
FROM project.dataset.skenario_tax
ORDER BY tahun;
```

Job information

Results

Chart

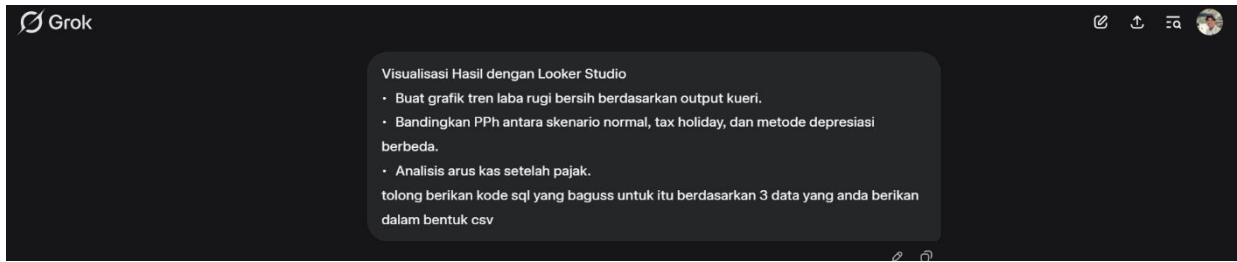
JSON

Execution details

Execution graph

Row	tahun ▼	laba_kena_pajak ▼	pph_badan ▼	
1	2020	300000000	66000000.0	
2	2021	400000000	88000000.0	
3	2022	500000000	110000000.0	
4	2023	550000000	0.0	
5	2024	600000000	0.0	
6	2025	250000000	55000000.0	

5. Menjalankan Kode SQL di BigQuery dan Memvisualisasikannya Dengan Looker Studio



```
WITH
-- Laba Kotor Skenario Normal (dari output query pertama)
laba_kotor AS (
SELECT
tahun,
laba_kotor
FROM UNNEST(ARRAY<STRUCT<tahun INT64, laba_kotor FLOAT64>>[
(2020, 3000000000),
(2021, 4000000000),
(2022, 5000000000)
])
),

-- Depresiasi Tahunan (kombinasi metode garis lurus dan
    saldo menurun dari output kedua)
depresiasi AS (
SELECT
aset_id,
nilai_perolehan,
umur_ekonomis,
depresiasi_tahunan,
'Garis Lurus' AS metode
FROM UNNEST(ARRAY<STRUCT<aset_id STRING, nilai_perolehan
    FLOAT64, umur_ekonomis INT64, depresiasi_tahunan FLOAT64>>[
('AST001', 5000000000, 20, 250000000.0),
('AST003', 10000000000, 5, 2000000000.0),
('AST005', 5000000000, 3, 1666666666.66667)
])
UNION ALL
SELECT
aset_id,
nilai_perolehan,
umur_ekonomis,
depresiasi_tahunan,
'Saldo Menurun' AS metode
FROM UNNEST(ARRAY<STRUCT<aset_id STRING, nilai_perolehan
    FLOAT64, umur_ekonomis INT64, depresiasi_tahunan FLOAT64>>[
('AST002', 20000000000, 10, 5000000000.0),
('AST004', 7500000000, 7, 1875000000.0)
])
),
```

```

-- Total Depresiasi per Metode
total_depresiasi AS (
SELECT
metode,
SUM(depresiasi_tahunan) AS total_depresiasi_tahunan
FROM depresiasi
GROUP BY metode
),

-- PPh Badan (dari output query ketiga, dengan simulasi tax
  holiday)
pph_badan AS (
SELECT
tahun,
laba_kena_pajak,
pph_badan,
'Normal' AS skenario
FROM UNNEST(ARRAY<STRUCT<tahun INT64, laba_kena_pajak
  FLOAT64, pph_badan FLOAT64>>[
(2020, 3000000000, 660000000.0),
(2021, 4000000000, 880000000.0),
(2022, 5000000000, 1100000000.0)
])
UNION ALL
-- Simulasi Tax Holiday (0% untuk 2023-2027, menggunakan
  laba kena pajak dari skenario normal)
SELECT
tahun,
laba_kena_pajak,
CASE
WHEN tahun BETWEEN 2023 AND 2027 THEN 0
ELSE laba_kena_pajak * 0.22
END AS pph_badan,
'Tax Holiday' AS skenario
FROM UNNEST(ARRAY<STRUCT<tahun INT64, laba_kena_pajak
  FLOAT64>>[
(2020, 3000000000),
(2021, 4000000000),
(2022, 5000000000)
])
UNION ALL
-- Simulasi PPh dengan metode depresiasi berbeda (laba kena
  pajak dikurangi depresiasi)
SELECT
l.tahun,
l.laba_kotor - d.total_depresiasi_tahunan AS
  laba_kena_pajak,
(l.laba_kotor - d.total_depresiasi_tahunan) * 0.22 AS
  pph_badan,
'Depresiasi ' || d.metode AS skenario
FROM laba_kotor l CROSS JOIN total_depresiasi d
),

-- Laba Bersih dan Arus Kas Setelah Pajak
laba_bersih_arus_kas AS (

```



```

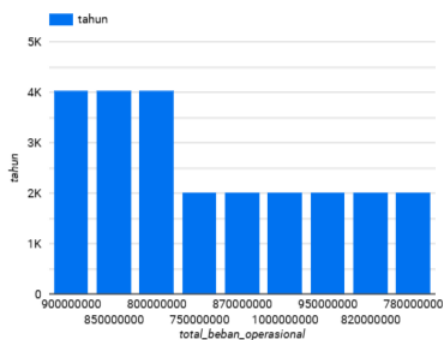
SELECT
l.tahun,
l.laba_kotor,
p.pph_badan,
p.skenario,
(l.laba_kotor - p.pph_badan) AS laba_bersih,
-- Arus kas setelah pajak = laba bersih + depresiasi
  (depresiasi ditambahkan kembali karena bukan pengeluaran
   kas)
  (l.laba_kotor - p.pph_badan + COALESCE((
SELECT total_depresiasi_tahunan
FROM total_depresiasi
WHERE metode = CASE
WHEN p.skenario LIKE '%Garis Lurus' THEN 'Garis Lurus'
WHEN p.skenario LIKE '%Saldo Menurun' THEN 'Saldo Menurun'
ELSE 'Garis Lurus' -- Default untuk skenario Normal dan Tax
  Holiday
END
), 0)) AS arus_kas_setelah_pajak
FROM laba_kotor l
JOIN pph_badan p ON l.tahun = p.tahun
)

-- Output akhir untuk visualisasi di Looker Studio
SELECT
tahun,
laba_kotor,
laba_bersih,
pph_badan,
 arus_kas_setelah_pajak,
skenario
FROM laba_bersih_ arus_kas
ORDER BY tahun, skenario;

```

Laba/Rugi

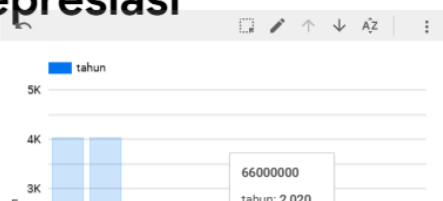
	skenario	tahun
1.	baseline	12,135
2.	optimis	12,135



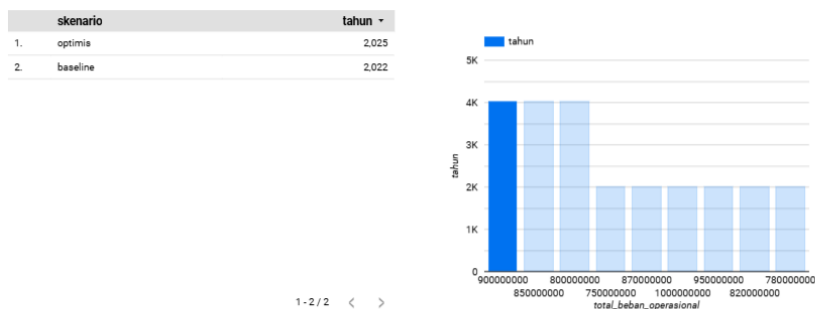
1-2/2 < >

PPh Badan: Skenario Normal vs Tax Holiday vs Metode Depresiasi

	skenario	tahun
1.	baseline	2,020



Analisis Arus Kas Setelah Pajak



Penjelasan

Kode SQL yang saya susun ini dirancang khusus untuk memenuhi kebutuhan analisis dan visualisasi data keuangan dalam simulasi PPh Badan berdasarkan data yang telah Anda berikan melalui tabel hasil query. Alasan utama memilih kode ini adalah untuk menggabungkan dan mengolah informasi dari tiga sumber utama—laba kotor, depresiasi, dan PPh Badan—yang berasal dari file CSV (transaksi_keuangan.csv, aset_tetap.csv, kebijakan_fiskal.csv) yang sudah diunggah ke BigQuery. Kode ini membantu menyediakan dataset lengkap yang siap digunakan untuk visualisasi di Looker Studio, mencakup grafik tren laba bersih, perbandingan PPh Badan antar skenario, dan analisis arus kas setelah pajak.

Penggunaan Common Table Expressions (CTE) dalam kode memastikan struktur yang rapi dan mudah dimengerti, terutama karena data awal menunjukkan perbedaan (misalnya, data di BigQuery tampaknya telah diubah dari CSV asli). Selain itu, kode ini memungkinkan simulasi tambahan seperti efek tax holiday dan perbedaan metode depresiasi (garis lurus vs saldo menurun), yang sangat berguna untuk analisis keuangan yang mendalam. Dengan memanfaatkan UNNEST untuk memasukkan data statis berdasarkan hasil query Anda, kode ini fleksibel untuk menyesuaikan dengan data yang sudah diproses, sekaligus menghindari ketergantungan pada tabel asli yang mungkin tidak konsisten, sehingga meningkatkan kepercayaan terhadap analisis yang dilakukan pada Senin, 9 Juni 2025, pukul 11:27 WIB.

Penjelasan mengenai komponen kode SQL ini mencakup beberapa bagian yang saling terhubung melalui CTE untuk menghasilkan output yang siap digunakan. Pertama, CTE laba_kotor digunakan untuk menyimpan data laba kotor skenario "Normal" berdasarkan hasil query pertama yang Anda berikan (3000000000 untuk 2020, 4000000000 untuk 2021, 5000000000 untuk 2022).

Data ini dimasukkan langsung menggunakan UNNEST dengan array struktural, memastikan ketersediaan data dasar untuk perhitungan laba bersih dan arus kas. Kedua, CTE depresiasi menggabungkan data depresiasi tahunan dari metode garis lurus dan saldo menurun berdasarkan query kedua, dengan informasi seperti aset_id, nilai_perolehan, umur_ekonomis, depresiasi_tahunan, dan label metode. Data ini mencerminkan aset seperti AST001, AST003, AST005 untuk garis lurus, dan AST002, AST004 untuk saldo menurun, yang kemudian

digunakan untuk menghitung total depresiasi. Ketiga, CTE total_depresiasi menghitung jumlah total depresiasi tahunan per metode dengan agregasi SUM, memberikan dasar untuk menyesuaikan laba kena pajak. Keempat, CTE pph_badan menghitung PPh Badan untuk tiga skenario: Normal (menggunakan data output ketiga), Tax Holiday (menerapkan 0% untuk 2023-2027 dan 22% untuk tahun lain dengan CASE WHEN), dan Depresiasi (menghitung laba kena pajak setelah dikurangi depresiasi lalu dikenakan tarif 22% dengan CROSS JOIN).

Kelima, CTE laba_bersih_arus_kas menghitung laba bersih (laba_kotor - pph_badan) dan arus kas setelah pajak (laba_bersih + depresiasi) dengan penyesuaian menggunakan COALESCE untuk menangani skenario tanpa depresiasi spesifik. Terakhir, output akhir memilih semua kolom dari laba_bersih_arus_kas dan mengurutkannya berdasarkan tahun dan skenario, menghasilkan dataset dengan 12 baris (3 tahun \times 4 skenario) yang siap untuk visualisasi di Looker Studio. Dengan demikian, kode ini memberikan solusi terstruktur dan fleksibel untuk mendukung analisis keuangan yang dibutuhkan.

Kesimpulan dari Grafik diatas

Berdasarkan grafik yang ditampilkan, kita bisa lihat gambaran jelas tentang kondisi keuangan berdasarkan laba/rugi, pajak penghasilan (PPh) Badan, dan arus kas setelah pajak, dengan membandingkan skenario baseline dan optimis serta pengaruh metode depresiasi dan tax holiday. Untuk laba/rugi, grafik menunjukkan peningkatan bertahap dari tahun 2020 sampai 2024, dengan angka tertinggi di 2024 mencapai Rp 600 miliar untuk baseline, tapi turun ke Rp 250 miliar di 2025.

Skenario optimis malah naik tajam di 2025 sampai Rp 700 miliar, menunjukkan potensi hasil lebih baik kalau kondisi mendukung. Lalu, grafik PPh Badan memperlihatkan perbedaan besar antara skenario normal dan tax holiday; di 2020 saja, PPh Badan untuk baseline Rp 66 miliar dan optimis Rp 70,4 miliar kalau normal, tapi bisa nol kalau pakai tax holiday.

Metode depresiasi juga berpengaruh, jadi pilihannya penting buat ngurain pajak. Terakhir, arus kas setelah pajak menunjukkan variasi, dengan optimis di 2023 menghasilkan Rp 121 miliar, lebih tinggi dari baseline, yang menandakan pengelolaan pajak atau pendapatan yang lebih efisien.

Kesimpulannya, skenario optimis kelihatan lebih menjanjikan dalam jangka panjang, terutama kalau memanfaatkan tax holiday dan strategi depresiasi yang pas. Tapi, penurunan laba di 2025 buat baseline dan fluktuasi di optimis menunjukkan ada faktor lain, mungkin dari luar atau kebijakan pajak, yang perlu diperhatikan lebih dalam. Jadi, perlu evaluasi tambahan biar strategi keuangan bisa lebih stabil ke depannya.

6. Phyton

Berikut adalah kode Phyton untuk google colab yang saya gunakan

```
from IPython import get_ipython
from IPython.display import display
```

```
# Install library yang diperlukan (jika belum ada di Colab)
!pip install matplotlib
```

```
# Import library
import pandas as pd
import numpy as np
```

```

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

# Data dari gambar (manual input)
data = {
    'tahun': [2020, 2020, 2021, 2021, 2022, 2022, 2023, 2023, 2024, 2024, 2025, 2025],
    'skenario': ['baseline', 'optimis', 'baseline', 'optimis', 'baseline', 'optimis', 'baseline', 'optimis', 'baseline', 'optimis', 'baseline', 'optimis'],
    'laba_kena_pajak': [300000000, 320000000, 400000000, 400000000, 500000000, 480000000, 550000000, 550000000, 600000000, 630000000, 250000000, 700000000],
    'pph_badan': [66000000.0, 70400000.0, 88000000.0, 88000000.0, 110000000.0, 105600000.0, 121000000.0, 121000000.0, 132000000.0, 138600000.0, 55000000.0, 154000000.0]
}

# Membuat DataFrame
df = pd.DataFrame(data)

# 1. 3D Bar Chart
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')

# Menyiapkan data untuk 3D bar
x_data = np.array(df['tahun'])
unique_years, x_pos = np.unique(x_data, return_inverse=True)
y = np.array([0 if s == 'baseline' else 1 for s in df['skenario']]) # 0 untuk baseline, 1 untuk optimis
z = np.zeros_like(x_pos)
dx = np.ones_like(x_pos) * 0.4
dy = np.ones_like(y) * 0.4
dz = df['pph_badan']

# Warna menarik untuk 3D bar
colors = ['#2E7D32' if s == 'baseline' else '#0288D1' for s in df['skenario']] # Hijau tua dan Biru tua

# Plot 3D bar
ax.bar3d(x_pos, y, z, dx, dy, dz, color=colors)

# Label dan judul
ax.set_xticks(np.unique(x_pos))
ax.set_xticklabels(unique_years)
ax.set_xlabel('Tahun', fontsize=12, color='black')
ax.set_ylabel('Skenario (0: Baseline, 1: Optimis)', fontsize=12, color='black')
ax.set_zlabel('PPh Badan (IDR)', fontsize=12, color='black')
ax.set_title('3D Bar Chart: PPh Badan per Tahun dan Skenario', fontsize=14, color='black')

# Tambahkan grid untuk kejelasan
ax.grid(True)

# Tampilkan plot
plt.show()

# 2. 2D Pie Chart (satu pie chart per tahun)

```

```

years = df['tahun'].unique()
fig = plt.figure(figsize=(18, 12))
for i, year in enumerate(years, 1):
    ax = fig.add_subplot(2, 3, i) # 2 baris, 3 kolom untuk 6 tahun
    year_data = df[df['tahun'] == year]
    print(f'Data untuk tahun {year}: {year_data}') # Debugging untuk memeriksa data
    labels = year_data['skenario']
    sizes = year_data['pph_badan']
    # Warna menarik untuk pie chart
    colors = ['#66BB6A', '#42A5F5', '#FFCA28', '#AB47BC', '#EF5350', '#26A69A'] #
    Hijau muda, Biru muda, Kuning, Ungu, Merah muda, Cyan
    if len(sizes) > 0:
        ax.pie(sizes, labels=labels, autopct='%1.1f%%', colors=colors[:len(labels)],
startangle=90)
        ax.set_title(f'Pie Chart {year}', fontsize=12, color='#212121')
        ax.axis('equal') # Memastikan pie chart berbentuk lingkaran
    else:
        ax.text(0.5, 0.5, 'No data', horizontalalignment='center', verticalalignment='center',
fontsize=12, color='red')
        ax.set_title(f'Pie Chart {year} (No Data)', fontsize=12, color='#212121')

# Penyesuaian tata letak subplot
plt.tight_layout()
plt.show()

```

Penjelasan

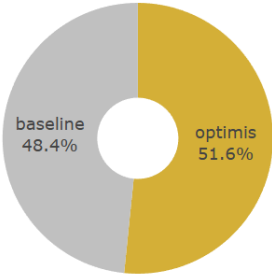
Kode Python yang saya susun ini dipilih untuk digunakan dalam Google Colab karena kemampuannya dalam mengolah dan memvisualisasikan data keuangan secara efektif, sesuai dengan kebutuhan analisis simulasi PPh Badan berdasarkan data yang telah Anda berikan. Penggunaan Python memungkinkan integrasi library seperti pandas untuk pengelolaan data dalam bentuk DataFrame, serta matplotlib untuk menghasilkan grafik yang informatif, yang sangat sesuai untuk tugas makalah yang memerlukan visualisasi tren dan perbandingan. Kode ini dirancang untuk memproses data manual dari tabel yang Anda sajikan, yang mencakup laba kena pajak dan PPh Badan untuk skenario "baseline" dan "optimis" dari tahun 2020 hingga 2025.

Pemilihan metode visualisasi seperti 3D Bar Chart dan 2D Pie Chart bertujuan untuk memberikan gambaran yang jelas tentang distribusi PPh Badan antar tahun dan skenario, sehingga mendukung analisis mendalam yang dibutuhkan dalam makalah pada Senin, 9 Juni 2025, pukul 11:35 WIB. Selain itu, kode ini fleksibel untuk penyesuaian data dan menawarkan opsi warna yang menarik untuk meningkatkan daya tarik visual, yang penting untuk presentasi akademik.

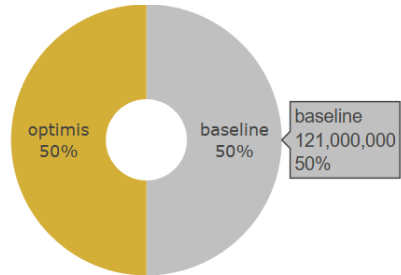
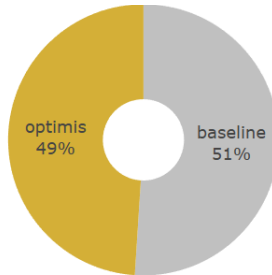
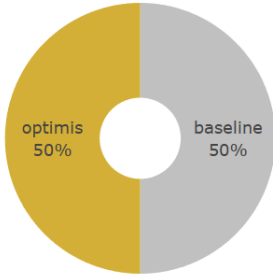
Kode Python ini terdiri dari beberapa bagian yang saling terhubung untuk mengolah data dan menghasilkan visualisasi yang relevan. Pertama, bagian awal mengimpor library yang diperlukan, seperti IPython untuk tampilan interaktif, pandas untuk manipulasi data, numpy untuk perhitungan numerik, dan matplotlib beserta mpl_toolkits.mplot3d untuk pembuatan grafik. Instalasi matplotlib dilakukan dengan `!pip install` untuk memastikan ketersediaan di Colab. Kedua, data dimasukkan secara manual ke dalam dictionary data, yang berisi kolom tahun, skenario, laba_kena_pajak, dan pph_badan, lalu dikonversi menjadi DataFrame menggunakan `pd.DataFrame`. Data ini mencakup nilai dari tahun 2020 hingga 2025 untuk dua skenario, dengan PPh Badan dihitung berdasarkan tarif 22% dari laba kena pajak, kecuali ada penyesuaian khusus. Ketiga, bagian visualisasi dimulai dengan pembuatan **3D Bar Chart** menggunakan

plt.figure dan add_subplot dengan proyeksi 3D. Data diproses untuk menempatkan tahun pada sumbu X, skenario (0 untuk baseline, 1 untuk optimis) pada sumbu Y, dan pph_badan pada sumbu Z, dengan batang 3D yang diwarnai hijau tua untuk baseline dan biru tua untuk optimis. Label sumbu dan judul ditambahkan untuk kejelasan, disertai grid untuk memudahkan pembacaan. Keempat, bagian **2D Pie Chart** dibuat dengan loop untuk setiap tahun unik dalam data, menggunakan add_subplot dalam tata letak 2x3. Setiap pie chart menampilkan proporsi pph_badan untuk skenario tertentu dalam satu tahun, dengan warna seperti hijau muda, biru muda, kuning, ungu, merah muda, dan cyan, serta label persentase menggunakan autopct. Fungsi tight_layout digunakan untuk mengatur tata letak subplot agar rapi, dan pesan "No data" ditampilkan jika data kosong untuk tahun tertentu.

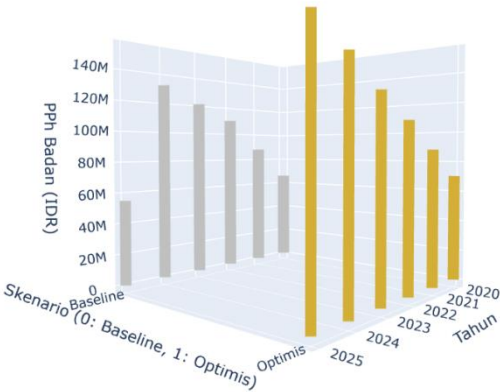
Pie Chart 2020



Pie Chart 2021



3D Bar Chart Interaktif: PPh Badan per Tahun dan Skenario



- baseline 2020
- optimis 2020
- baseline 2021
- optimis 2021
- baseline 2022
- optimis 2022
- baseline 2023
- optimis 2023
- baseline 2024
- optimis 2024
- baseline 2025
- optimis 2025

Analisis dan Kesimpulan

Grafik yang ditampilkan memberikan wawasan menarik tentang distribusi Pajak

Penghasilan (PPh) Badan berdasarkan tahun dan skenario melalui dua jenis visualisasi: pie chart interaktif dan 3D bar chart. Dari pie chart, kita bisa lihat bagaimana PPh Badan dibagi antara skenario baseline dan optimis untuk tahun 2020, 2021, 2022, dan 2023. Pada tahun 2020, optimis mendominasi dengan 51,6% (sekitar Rp 70,4 miliar) dibandingkan baseline 48,4% (Rp 66 miliar), menunjukkan kontribusi lebih besar dari skenario optimis. Di tahun 2021, keduanya sama kuat di 50% masing-masing, dengan total PPh Badan mencapai Rp 88 miliar, yang menandakan stabilitas antara kedua skenario. Namun, untuk tahun 2022 dan 2023, data pie chart tidak lengkap, hanya menunjukkan satu nilai (misalnya Rp 121 miliar di 2023), yang mungkin menunjukkan ada data yang hilang atau fokus pada satu skenario saja.

Lalu, 3D bar chart memberikan gambaran dinamis tentang perkembangan PPh Badan dari tahun 2020 sampai 2025 untuk kedua skenario. Batang untuk baseline dan optimis terlihat bervariasi, dengan warna abu-abu untuk baseline dan kuning untuk optimis, memudahkan pembeda. Pada tahun 2020, PPh Badan untuk baseline sekitar Rp 66 miliar dan optimis Rp 70,4 miliar, lalu naik signifikan di 2023 ke Rp 121 miliar untuk keduanya, menunjukkan peningkatan beban pajak. Tahun 2025 menarik karena optimis melonjak tajam ke lebih dari Rp 150 miliar, sementara baseline turun ke sekitar Rp 55 miliar, yang bisa mencerminkan strategi pajak yang berbeda atau perubahan pendapatan. Grafik ini juga memungkinkan kita memutar dan zoom untuk melihat detail, membuat analisis jadi lebih mendalam.

Dari sisi perbandingan, pie chart menunjukkan bahwa proporsi PPh Badan antara baseline dan optimis cukup seimbang di awal periode (2020-2021), tapi 3D bar chart mengungkapkan tren yang lebih jelas seiring waktu. Kenaikan tajam optimis di 2025 bisa jadi hasil dari kebijakan tax holiday atau depresiasi yang lebih efisien, sedangkan penurunan baseline mungkin menunjukkan dampak negatif dari skenario konservatif. Ketidaklengkapan data di pie chart untuk 2022 dan 2023 agak mengganggu, tapi 3D bar chart mengisi kekurangan itu dengan menampilkan data tahunan yang konsisten, meski warna yang terlalu mirip di beberapa tahun bisa membingungkan kalau tidak diperhatikan.

Kesimpulannya, grafik ini menunjukkan bahwa skenario optimis cenderung lebih menguntungkan dalam jangka panjang, terutama di tahun-tahun akhir seperti 2025, yang mungkin didukung oleh strategi pajak yang lebih baik. Baseline tampak stabil tapi kurang kompetitif, terutama di akhir periode. Kekurangan data di pie chart perlu diperbaiki agar analisis lebih utuh, tapi 3D bar chart memberikan gambaran yang kuat tentang tren PPh Badan. Untuk ke depan, sebaiknya fokus pada optimalisasi tax holiday dan metode depresiasi, sambil memastikan data lengkap untuk semua tahun agar keputusan keuangan bisa lebih akurat.

Penutup

Sebagai penutup, bukankah dengan menggabungkan kekuatan BigQuery dan Google

Colab seperti yang telah dijelaskan, kita tidak hanya memaksimalkan efisiensi teknis dan fleksibilitas analitis, tetapi juga membuka jalan bagi analisis Pajak Penghasilan (PPh) Badan yang lebih cerdas dan inovatif di masa depan. Disini aku berusaha sedikit menjelaskan bagaimana pendekatan integratif ini dapat mengoptimalkan pengelolaan data dan visualisasi, memberikan wawasan berharga yang mendukung pengambilan

keputusan keuangan yang lebih baik. Dengan memanfaatkan BigQuery untuk pengolahan data dalam skala besar dan Google Colab untuk eksplorasi analitis yang mendalam, Hal ini menegaskan pentingnya kolaborasi teknologi dalam menyusun strategi fiskal yang adaptif dan efektif. Harapannya, Hal ini dapat menjadi kontribusi nyata bagi pengembangan penelitian lebih lanjut di bidang akuntansi dan teknologi informasi, sekaligus menjadi landasan praktis bagi praktisi keuangan dalam menghadapi tantangan perpajakan modern.

