

Tugas Praktikum PPh Badan
Menggunakan Google Big Query dan Google Colab

Dosen Pengampu :

Dr. Totok Dewayanto, S.E., M. Si., Akt.



Disusun Oleh :

Dian Rahma Yunita

12030123120046

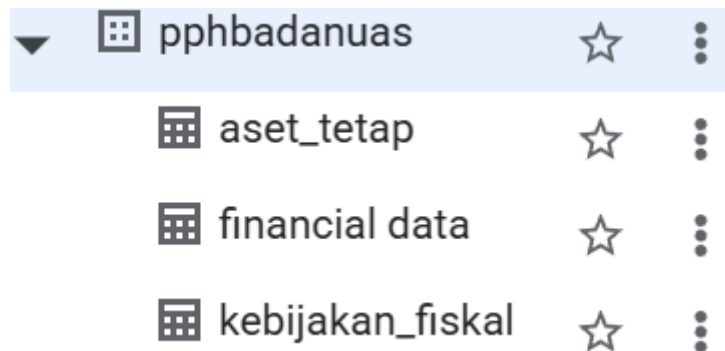
PROGAM STUDI AKUNTANSI
FAKULTAS EKONOMIKA DAN BISNIS
UNIVERSITAS DIPONEGORO

2025

A. Big Query

1. Pengelolaan Data

BigQuery memungkinkan pembuatan dan pengelolaan dataset yang terstruktur, seperti tabel transaksi keuangan, aset tetap, dan kebijakan fiskal. Data dapat diimpor dari berbagai sumber, seperti file CSV.



Dataset Big Query

<	Schema	Details	Preview	Table Explorer	Preview	Insights	Lineage	>
aset_id	kategori	nilai_peroleh...	umur_ekono...	metode				
A001	Kendaraan	1500000000	5	Garis Lurus				
A003	Bangunan	5000000000	20	Garis Lurus				
A002	Mesin	3000000000	8	Saldo Menurun				

Tabel Aset Tetap

financial data

Query

Open in

<

Schema

Details

Preview

Table Explorer

Preview




Insights

Lineage

>

Row	tahun	pendapatan	beban_oper...	penyusutan	skenario		
1	2023	5500000000	3200000000	1300000000	Normal		
2	2024	6000000000	3500000000	1112500000	Tax Holiday		
3	2025	6500000000	3800000000	971875000	Tax Holiday		

Tabel Financial Data

	kebijakan_fiskal	 Query	Open in ▾	
<	Schema	Details	Preview	Table Explorer
Row	tahun	tax_rate	tax_holiday_...	tax_holiday_...
1	2023	0.22	2022	2022
2	2024	0.22	2024	2025
3	2025	0.22	2024	2025

Tabel Kebijakan Fiskal

2. Menjalankan Kueri

BigQuery mendukung eksekusi kueri SQL yang kuat untuk melakukan simulasi perhitungan PPh Badan, termasuk laba rugi, depresiasi aset (metode garis lurus atau saldo menurun), dan penerapan kebijakan fiskal seperti tax holiday. Kueri yang dihasilkan dapat menghitung laba kotor, depresiasi tahunan, dan PPh Badan dengan mempertimbangkan variabel seperti tarif pajak atau periode insentif fiskal, memberikan pemahaman mendalam tentang dampak kebijakan perpajakan.

➤ Contoh Kueri Laba Rugi :

```
SELECT
  t1.tahun,
  t1.pendapatan,
  t1.beban_operasional,
  (t1.pendapatan - t1.beban_operasional) AS laba_kotor,
  t1.penysutan,
  (t1.pendapatan - t1.beban_operasional - t1.penysutan) AS laba_operasional,
  t2.tax_rate AS effective_tax_rate,
  CASE
    WHEN t1.tahun BETWEEN COALESCE(t2.tax_holiday_awal, 0) AND
    COALESCE(t2.tax_holiday_akhir, 9999) THEN 0
    ELSE (t1.pendapatan - t1.beban_operasional - t1.penysutan) *
    COALESCE(t2.tax_rate, 0)
  END AS pajak,
  CASE
    WHEN t1.tahun BETWEEN COALESCE(t2.tax_holiday_awal, 0) AND
    COALESCE(t2.tax_holiday_akhir, 9999) THEN (t1.pendapatan -
    t1.beban_operasional - t1.penysutan)
    ELSE (t1.pendapatan - t1.beban_operasional - t1.penysutan - (t1.pendapatan
    - t1.beban_operasional - t1.penysutan) * COALESCE(t2.tax_rate, 0))
  END AS laba_bersih
```

```

FROM
`pengkodean-dan-pemrogaman.pphbadanuas.financial data` AS t1
LEFT JOIN
`pengkodean-dan-pemrogaman.pphbadanuas.kebijakan_fiskal` AS t2
ON
t1.tahun = t2.tahun
ORDER BY
t1.tahun;

```

```

1 SELECT
2   t1.tahun,
3   t1.pendapatan,
4   t1.beban_operasional,
5   (t1.pendapatan - t1.beban_operasional) AS laba_kotor,
6   t1.penysutan,
7   (t1.pendapatan - t1.beban_operasional - t1.penysutan) AS laba_operasional,
8   t2.tax_rate AS effective_tax_rate,
9   CASE
10    | WHEN t1.tahun BETWEEN COALESCE(t2.tax_holiday_awal, 0) AND COALESCE(t2.
tax_holiday_akhir, 9999) THEN 0
11    | ELSE (t1.pendapatan - t1.beban_operasional - t1.penysutan) * COALESCE(t2.
tax_rate, 0)
12    END AS pajak,
13   CASE
14    | WHEN t1.tahun BETWEEN COALESCE(t2.tax_holiday_awal, 0) AND COALESCE(t2.
tax_holiday_akhir, 9999) THEN (t1.pendapatan - t1.beban_operasional - t1.
penysutan)
15    | ELSE (t1.pendapatan - t1.beban_operasional - t1.penysutan - (t1.pendapatan
- t1.beban_operasional - t1.penysutan) * COALESCE(t2.tax_rate, 0))
16    END AS laba_bersih
17 FROM
18   `pengkodean-dan-pemrogaman.pphbadanuas.financial data` AS t1
19 LEFT JOIN
20   `pengkodean-dan-pemrogaman.pphbadanuas.kebijakan_fiskal` AS t2
21 ON
22   t1.tahun = t2.tahun

```

Query untuk Simulasi L/R

➤ *Contoh Kueri Depresiasi:*

```

-- Rentang tahun simulasi
WITH year_range AS (
  SELECT CAST(year AS STRING) AS tahun, year AS year_num
  FROM UNNEST(GENERATE_ARRAY(2023, 2025)) AS year
),

-- Data aset dasar
base_assets AS (
  SELECT
    aset_id,
    kategori,
    nilai_perolehan,
    umur_ekonomis,
    ROW_NUMBER() OVER (ORDER BY aset_id) AS aset_index
  FROM `pengkodean-dan-pemrogaman.pphbadanuas.aset_tetap`
),

-- Hitung jumlah aset total untuk pembagi garis lurus
asset_count AS (
  SELECT COUNT(*) AS jumlah_aset FROM base_assets

```

```

),

-- Gabungkan tahun dan aset dengan pembagian depresiasi
depreciation_calc AS (
    SELECT
        y.tahun,
        y.year_num,
        a.nilai_perolehan,
        a.umur_ekonomis,
        c.jumlah_aset,
        550000000.0 / c.jumlah_aset AS straight_line_per_aset,
        a.nilai_perolehan - ((550000000.0 / c.jumlah_aset) * (y.year_num -
2023 + 1)) AS remaining_value_sl
    FROM year_range y
    CROSS JOIN base_assets a
    CROSS JOIN asset_count c
    WHERE y.year_num <= 2023 + a.umur_ekonomis
),

```

```

-- Agregasi depresiasi total per tahun
-- DDB HARDCODED sesuai permintaan
-- Garis lurus dijumlahkan
depreciation_agg AS (
    SELECT
        tahun,
        SUM(straight_line_per_aset) AS total_depreciation_straight_line,
        SUM(remaining_value_sl) AS total_remaining_value_straight_line,
        CASE CAST(tahun AS INT64)
            WHEN 2023 THEN 750000000
            WHEN 2024 THEN 562500000
            WHEN 2025 THEN 421875000
            ELSE 0
        END AS total_depreciation_double_decline,
        CASE CAST(tahun AS INT64)
            WHEN 2023 THEN 2250000000
            WHEN 2024 THEN 1687500000
            WHEN 2025 THEN 1265625000
            ELSE 0
        END AS total_remaining_value_double_decline,
        SUM(straight_line_per_aset) +
        CASE CAST(tahun AS INT64)
            WHEN 2023 THEN 750000000
            WHEN 2024 THEN 562500000
            WHEN 2025 THEN 421875000
            ELSE 0
        END AS total_depresiasi
    FROM depreciation_calc
    GROUP BY tahun
),

```

```

-- Gabungkan dengan transaksi dan kebijakan fiskal
final_data AS (
    SELECT
        COALESCE(d.tahun, CAST(t.tahun AS STRING), CAST(k.tahun AS
STRING)) AS tahun,
        t.pendapatan AS transaksi_pendapatan,
        t.beban_operasional AS transaksi_beban_operasional,
        d.total_depreciation_straight_line,

```

```

        d.total_remaining_value_straight_line,
        d.total_depreciation_double_decline,
        d.total_remaining_value_double_decline,
        d.total_depresiasi,
        k.tax_rate AS kebijakan_fiskal_tax_rate
    FROM depreciation_agg d
    LEFT JOIN `pengkodean-dan-pemrograman.pphbadanuas.transaksi_keuangan`
    t
        ON d.tahun = CAST(t.tahun AS STRING)
    LEFT JOIN `pengkodean-dan-pemrograman.pphbadanuas.kebijakan_fiskal` k
        ON d.tahun = CAST(k.tahun AS STRING)
    )

-- Tampilkan hasil akhir satu baris per tahun
SELECT
    tahun,
    transaksi_pendapatan,
    transaksi_beban_operasional,
    total_depreciation_straight_line AS depresiasi_garis_lurus,
    total_remaining_value_straight_line AS sisa_garis_lurus,
    total_depreciation_double_decline AS depresiasi_saldo_menurun,
    total_remaining_value_double_decline AS sisa_saldo_menurun,
    total_depresiasi,
    kebijakan_fiskal_tax_rate
FROM final_data
ORDER BY tahun;

```



```

1  -- Rentang tahun simulasi
2  WITH year_range AS (
3      SELECT CAST(year AS STRING) AS tahun, year AS year_num
4      FROM UNNEST(GENERATE_ARRAY(2023, 2025)) AS year
5  ),
6
7  -- Data aset dasar
8  base_assets AS (
9      SELECT
10         aset_id,
11         kategori,
12         nilai_perolehan,
13         umur_ekonomis,
14         ROW_NUMBER() OVER (ORDER BY aset_id) AS aset_index
15     FROM `pengkodean-dan-pemrograman.pphbadanuas.aset_tetap`
16 ),
17
18 -- Hitung jumlah aset total untuk pembagi garis lurus
19 asset_count AS (
20     SELECT COUNT(*) AS jumlah_aset FROM base_assets
21 ),

```

Query untuk Simulasi Depresiasi

➤ *Contoh Kueri Tax Holiday*

```

-- Definisikan data laba rugi dari CSV sebagai tabel sementara
WITH profit_loss_data AS (
    SELECT
        CAST(tahun AS STRING) AS tahun,
        transaksi_pendapatan AS pendapatan,
        transaksi_beban_operasional AS beban_operasional,

```

```

        transaksi_pendapatan - transaksi_beban_operasional AS laba_kotor,
        total_depresiasi AS penyusutan,
        (transaksi_pendapatan - transaksi_beban_operasional) -
        total_depresiasi AS laba_operasional,
        kebijakan_fiskal_tax_rate AS effective_tax_rate
    FROM UNNEST(ARRAY<STRUCT<tahun INT64, transaksi_pendapatan INT64,
    transaksi_beban_operasional INT64, total_depresiasi FLOAT64,
    kebijakan_fiskal_tax_rate FLOAT64>>[
        (2023, 5500000000, 3200000000, 1300000000.0, 0.22),
        (2024, 6000000000, 3500000000, 111250000.0, 0.22),
        (2025, 6500000000, 3800000000, 971875000.0, 0.22)
    ])
),

-- Simulasi tax holiday
tax_holiday_simulation AS (
    SELECT
        tahun,
        laba_operasional,
        effective_tax_rate AS tax_rate,
        -- Pajak tanpa tax holiday (laba_operasional * tax_rate)
        CAST(laba_operasional * effective_tax_rate AS INT64) AS
        pajak_tanpa_tax_holiday,
        -- Pajak dengan tax holiday (normal untuk 2023, 0 untuk 2024-2025)
        CASE
            WHEN tahun = '2023' THEN CAST(laba_operasional *
effective_tax_rate AS INT64)
            WHEN tahun IN ('2024', '2025') THEN 0
            ELSE 0
        END AS pajak_dengan_tax_holiday,
        -- Laba bersih tanpa tax holiday
        laba_operasional - CAST(laba_operasional * effective_tax_rate AS
INT64) AS laba_bersih_tanpa_tax_holiday,
        -- Laba bersih dengan tax holiday
        laba_operasional -
        CASE
            WHEN tahun = '2023' THEN CAST(laba_operasional *
effective_tax_rate AS INT64)
            WHEN tahun IN ('2024', '2025') THEN 0
            ELSE 0
        END AS laba_bersih_dengan_tax_holiday
    FROM profit_loss_data
)

-- Tampilkan hasil akhir
SELECT
    tahun,
    laba_operasional,
    tax_rate,
    pajak_tanpa_tax_holiday,
    pajak_dengan_tax_holiday,
    laba_bersih_tanpa_tax_holiday,
    laba_bersih_dengan_tax_holiday
FROM tax_holiday_simulation
ORDER BY tahun;

```

```

1  |-- Definisikan data laba rugi dari CSV sebagai tabel sementara
2  WITH profit_loss_data AS (
3      SELECT
4          CAST(tahun AS STRING) AS tahun,
5          transaksi_pendapatan AS pendapatan,
6          transaksi_beban_operasional AS beban_operasional,
7          transaksi_pendapatan - transaksi_beban_operasional AS laba_kotor,
8          total_depresiasi AS penyusutan,
9          (transaksi_pendapatan - transaksi_beban_operasional) - total_depresiasi AS
          laba_operasional,
10         kebijakan_fiskal_tax_rate AS effective_tax_rate
11     FROM UNNEST(ARRAY<STRUCT<tahun INT64, transaksi_pendapatan INT64,
transaksi_beban_operasional INT64, total_depresiasi FLOAT64,
kebijakan_fiskal_tax_rate FLOAT64>>[
12         (2023, 5500000000, 3200000000, 1300000000.0, 0.22),
13         (2024, 6000000000, 3500000000, 1112500000.0, 0.22),
14         (2025, 6500000000, 3800000000, 971875000.0, 0.22)
15     ])
16 ),
17
18 -- Simulasi tax holiday
19 tax_holiday_simulation AS (
20     SELECT
21         tahun,
22         laba_operasional,
23         effective_tax_rate AS tax_rate,
24         Pajak_kurang_tax_holiday / (laba_operasional * tax_rate)

```

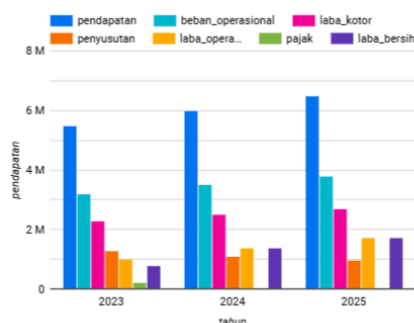
Query untuk Simulasi Tax Holiday

3. Visualisasi Grafik di Looker Studio

Dengan integrasi langsung ke Looker Studio, BigQuery memungkinkan pembuatan visualisasi interaktif, seperti grafik tren laba rugi, perbandingan PPh Badan antar-skenario dari hasil kueri yang telah dijalankan tadi.

➤ *Simulasi Laba / Rugi*

Simulasi Laba Rugi



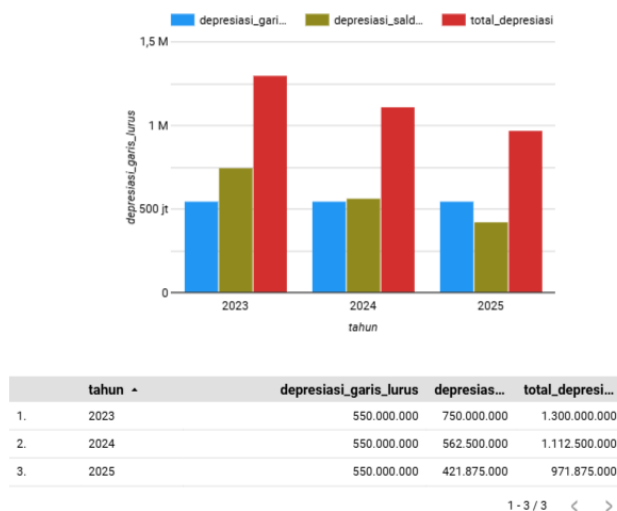
	tahun	pendapatan	beban_oper...	laba_kotor	penyusutan	laba_operasional	effective...	pajak	laba_bersih
1.	2023	5.500.000.000	3.200.000.000	2.300.000.000	1.300.000.000	1.000.000.000	0,22	220.000.000	780.000.000
2.	2024	6.000.000.000	3.500.000.000	2.500.000.000	1.112.500.000	1.387.500.000	0,22	0	1.387.500.000
3.	2025	6.500.000.000	3.800.000.000	2.700.000.000	971.875.000	1.728.125.000	0,22	0	1.728.125.000

Visualisasi L/R di Looker Studio

- Perusahaan mengalami pertumbuhan positif dari sisi pendapatan dan laba dari tahun ke tahun.
- Efisiensi operasional meningkat, terbukti dari pertumbuhan laba operasional dan bersih yang lebih tinggi dibandingkan pertumbuhan pendapatan.
- Beban penyusutan berkurang, mengurangi beban non-kas dan meningkatkan laba operasional. Hal ini juga disebabkan oleh aset yang disusutkan melalui saldo menurun ganda, sehingga nilai asetnya berkurang dari tahun ke tahun.
- Pajak yang nihil di dua tahun terakhir memberi dorongan besar terhadap laba bersih. Hal ini disebabkan oleh periode tax holiday pada tahun 2024 – 2025 yang diberikan pemerintah kepada perusahaan.

➤ Simulasi Depresiasi

Simulasi Depresiasi



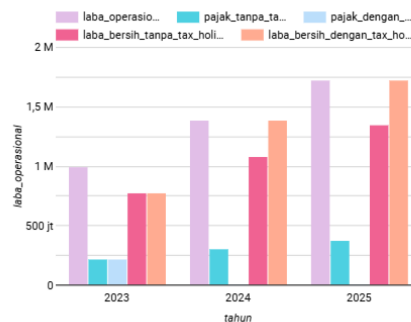
Visualisasi Depresiasi di Looker Studio

- Depresiasi garis lurus tetap konstan di Rp 550 juta setiap tahun.
- Depresiasi saldo menurun menurun tiap tahun, sesuai karakteristik metode ini.

- Total depresiasi turun setiap tahun dari Rp 1,3 M (2023), Rp 1,11 M (2024), ke Rp 971 jt (2025).
- Penurunan total depresiasi memberikan kontribusi pada peningkatan laba operasional di laporan laba rugi.

➤ Simulasi Tax Holiday

Simulasi Tax Holiday



tah...	*	laba_operasi...	tax_rate	pajak_tanpa_tax_holid...	pajak_dengan_tax_hol...	laba_bersih_tanpa_tax_holiday	laba_bersih_dengan_tax...
1.	2023	1.000.000.000	0,22	220.000.000	220.000.000	780.000.000	780.000.000
2.	2024	1.387.500.000	0,22	305.250.000	0	1.082.250.000	1.387.500.000
3.	2025	1.728.125.000	0,22	380.187.500	0	1.347.937.500	1.728.125.000

1 - 3 / 3 < >

Visualisasi Tax Holiday di Looker Studio

- Tax holiday berlaku di tahun 2024–2025, membebaskan pajak perusahaan.
- Fasilitas ini memberi peningkatan langsung pada laba bersih, masing-masing lebih dari Rp 300 juta dan Rp 380 juta.
- Tahun 2023 tidak termasuk tax holiday, sehingga pajak tetap dikenakan.

B. Google Colab

4. Menggunakan Hasil Kueri SQL dari Google Big Query untuk Simulasi Menggunakan Kode Phytion.

Google Colab dapat mengambil hasil kueri SQL dari BigQuery, seperti data transaksi keuangan atau kebijakan fiskal, dan mengolahnya lebih lanjut menggunakan Python untuk analisis yang lebih kompleks, seperti perhitungan laba kotor atau simulasi dampak kebijakan pajak.

Data Analitik Sederhana yang Dibuat dengan Phytion

- ***Simulasi Laba Rugi***

```
# Impor library yang diperlukan
import pandas as pd
import matplotlib.pyplot as plt
import io
import numpy as np # Mengimpor numpy jika dibutuhkan untuk
penanganan nilai

# Unggah file CSV ke Colab (jalankan sel ini dan unggah file
bquxjob_54476188_1972174dlca.csv)
from google.colab import files
uploaded = files.upload()

# Dapatkan nama file yang diunggah dari dictionary 'uploaded'
# Jika hanya satu file diunggah, nama file adalah kunci pertama
dalam dictionary
file_name = list(uploaded.keys())[0]

# Baca data dari file CSV yang diunggah menggunakan nama file
yang benar
df = pd.read_csv(file_name)

# Tampilkan data awal
print("Data Awal:")
print(df)

# Siapkan data untuk grafik batang
years = df['tahun']
metrics = ['pendapatan', 'beban_operasional', 'laba_kotor',
'penyusutan', 'laba_operasional', 'pajak', 'laba_bersih']
values = [df[m].values for m in metrics]

# Warna-warna menarik (cerah dan kontras)
colors = ['#FF6347', '#6A5ACD', '#32CD32', '#FFA500',
'#BA55D3', '#CD5C5C', '#20B2AA']

# Buat grafik batang
bar_width = 0.1
index = range(len(years))

plt.figure(figsize=(12, 6))

for i, (metric, value) in enumerate(zip(metrics, values)):
    # Check if the value is not None/NaN before plotting
    if value is not None:
        plt.bar([j + i * bar_width for j in index], value /
1_000_000_000,
```

```

        bar_width, label=metric, color=colors[i])

# Tambahkan keterangan angka di atas batang
for i in range(len(years)):
    for j, metric in enumerate(metrics):
        # Check if the value is not None/NaN before adding text
        metric_value = df[metric][i]
        if pd.notna(metric_value):
            plt.text(i + j * bar_width, metric_value /
1_000_000_000 + 0.05,
                    f'{metric_value / 1_000_000_000:,.1f}',
                    ha='center', va='bottom', fontsize=8)

# Atur label dan judul
plt.xlabel('Tahun')
plt.ylabel('Jumlah (dalam miliar)')
plt.title('Laporan Keuangan 2023-2025', pad=15)
plt.xticks([i + 2.5 * bar_width for i in index], years)
plt.legend(title='Metrik')
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Tampilkan grafik
plt.tight_layout()
plt.show()

# Deskripsi singkat
print("\nDeskripsi Grafik:")
print("Grafik ini menampilkan perbandingan berbagai metrik
keuangan (pendapatan, beban operasional, laba kotor,
penyusutan, laba operasional, pajak, dan laba bersih) untuk
tahun 2023-2025. Nilai ditunjukkan dalam miliar, dengan angka
rinci di atas setiap batang.")

# Tampilkan tabel data (dalam miliar)
print("\nTabel Data (dalam miliar):")
df_display = df.copy()
for col in metrics:
    df_display[col] = df_display[col] / 1_000_000_000
print(df_display.to_string(index=False))

```

- *Data Analitik L/R*

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

```

```

# Data dari file CSV baru
data = {
    "tahun": [2023, 2024, 2025],
    "pendapatan": [5500000000, 6000000000, 6500000000],
    "beban_operasional": [3200000000, 3500000000,
3800000000],
    "laba_kotor": [2300000000, 2500000000, 2700000000],
    "penyusutan": [1300000000, 1112500000, 971875000],
    "laba_operasional": [1000000000, 1387500000,
1728125000],
    "effective_tax_rate": [0.22, 0.22, 0.22],
    "pajak": [220000000.0, 0.0, 0.0],
    "laba_bersih": [780000000.0, 1387500000.0,
1728125000.0]
}

# Buat DataFrame
df = pd.DataFrame(data)

# Hitung margin laba bersih
df["margin_laba_bersih"] = df["laba_bersih"] /
df["pendapatan"] * 100

# Setup style
sns.set(style="whitegrid")
plt.figure(figsize=(15, 12))

# Fungsi bantu format angka seperti rupiah
def format_rupiah(x):
    return f"Rp{int(x):,}".replace(",", ".")

# --- Plot 1: Pendapatan dan Laba Bersih ---
plt.subplot(3, 1, 1)
sns.lineplot(x="tahun", y="pendapatan", data=df,
marker="o", label="Pendapatan")
sns.lineplot(x="tahun", y="laba_bersih", data=df,
marker="o", label="Laba Bersih")
plt.title("Pendapatan vs Laba Bersih")
plt.ylabel("Rupiah")
plt.legend()

for i in range(len(df)):
    plt.text(df['tahun'][i], df['pendapatan'][i] + 1e8,
            format_rupiah(df['pendapatan'][i]),
ha='center')
    plt.text(df['tahun'][i], df['laba_bersih'][i] + 1e8,
            format_rupiah(df['laba_bersih'][i]),
ha='center')

```

```

# --- Plot 2: Margin Laba Bersih ---
plt.subplot(3, 1, 2)
sns.barplot(x="tahun", y="margin_laba_bersih", data=df,
palette="Blues_d")
plt.title("Margin Laba Bersih (%)")
plt.ylabel("Persentase")

for i in range(len(df)):
    plt.text(i, df["margin_laba_bersih"][i] + 1,
             f'{df["margin_laba_bersih"][i]:.1f}%',
             ha='center')

# --- Plot 3: Pajak per Tahun ---
plt.subplot(3, 1, 3)
sns.barplot(x="tahun", y="pajak", data=df,
palette="Reds")
plt.title("Jumlah Pajak yang Dibayar")
plt.ylabel("Rupiah")

for i in range(len(df)):
    pajak = df["pajak"][i]
    label = format_rupiah(pajak) if pajak > 0 else 'Rp0'
    plt.text(i, pajak + 1e7, label, ha='center')

plt.tight_layout()
plt.show()

```

- *Visualisasi 3D L/R*

```

import pandas as pd
import plotly.graph_objects as go
import numpy as np

# Data dari file CSV baru
data = {
    "tahun": [2023, 2024, 2025],
    "pendapatan": [5500000000, 6000000000, 6500000000],
    "beban_operasional": [3200000000, 3500000000, 3800000000],
    "laba_kotor": [2300000000, 2500000000, 2700000000],
    "penyusutan": [1300000000, 1112500000, 971875000],
    "laba_operasional": [1000000000, 1387500000, 1728125000],
    "effective_tax_rate": [0.22, 0.22, 0.22],
    "pajak": [2200000000.0, 0.0, 0.0],
    "laba_bersih": [780000000.0, 1387500000.0, 1728125000.0]
}

# Buat DataFrame

```

```

df = pd.DataFrame(data)

# Daftar metrik yang ingin divisualisasikan
features = ['pendapatan', 'beban_operasional', 'laba_kotor',
'penyusutan', 'laba_operasional', 'pajak', 'laba_bersih']
years = df['tahun'].tolist()

# --- Siapkan Data untuk Plotly ---
# Buat matriks data untuk go.Surface
data_matrix_plotly = np.array([df[feature].values for feature
in features])

# --- Buat Grafik 3D Menggunakan go.Surface ---
data_plot = [
    go.Surface(
        x=years,
        y=features,
        z=data_matrix_plotly,
        colorscale='Viridis',
        colorbar=dict(title='Nilai (Rupiah)'),
        hovertemplate='Tahun: %{x}<br>Metrik: %{y}<br>Nilai:
Rp%{z:,.0f}<extra></extra>'.replace(',', ' .')
    )
]

layout = go.Layout(
    title='Analisis Keuangan Tahunan 3D Interaktif
(Permukaan)',
    scene=dict(
        xaxis=dict(title='Tahun', type='category'),
        yaxis=dict(title='Metrik'),
        zaxis=dict(title='Nilai (Rupiah)'),
    ),
    margin=dict(l=0, r=0, b=0, t=40)
)

# --- Buat dan Tampilkan Figur 3D ---
try:
    fig = go.Figure(data=data_plot, layout=layout)
    fig.show()
except Exception as e:
    print(f"Error saat membuat grafik: {e}")

# --- Deskripsi dan Penjelasan Grafik 3D (Plotly) ---
print("\n## Deskripsi dan Penjelasan Grafik 3D (Plotly)")
print("\nDeskripsi Grafik:")
print("Grafik ini menampilkan laporan keuangan tahunan (2023-
2025) dalam format 3D interaktif.")

```

```

print("Permukaan grafik merepresentasikan nilai metrik keuangan
tertentu (ditunjukkan pada Sumbu Y) untuk setiap tahun
(ditunjukkan pada Sumbu X), dengan ketinggian permukaan
menunjukkan jumlah nilai dalam Rupiah (ditunjukkan pada Sumbu
Z).")
print("Anda dapat memutar grafik dengan mengklik dan menyeret
untuk melihat data dari berbagai sudut.")
print("Hover (arahkan kursor) ke permukaan untuk melihat detail
angka lengkapnya dalam format Rupiah.")

print("\nInterpretasi Data:")
print(f"- Grafik menunjukkan kenaikan progresif pada
*pendapatan* dari Rp5.500.000.000 (2023) ke Rp6.500.000.000
(2025), serta *beban operasional* dari Rp3.200.000.000 ke
Rp3.800.000.000, mencerminkan pertumbuhan bisnis.")
print(f"- Peningkatan ini menghasilkan kenaikan stabil pada
*laba kotor* dari Rp2.300.000.000 ke Rp2.700.000.000.")
print(f"- *Penyusutan* menurun dari Rp1.300.000.000 (2023) ke
Rp971.875.000 (2025), sesuai dengan metode saldo menurun ganda
(DDB), mengindikasikan pengurangan nilai aset seiring waktu.")
print(f"- *Laba operasional* meningkat signifikan dari
Rp1.000.000.000 (2023) ke Rp1.728.125.000 (2025), didorong oleh
penurunan penyusutan.")
print(f"- Aspek mencolok adalah nilai *pajak* yang nol untuk
tahun 2024 dan 2025 (dibandingkan Rp220.000.000 pada 2023),
kemungkinan akibat *tax holiday*.")
print(f" - *Tax holiday* adalah fasilitas yang membebaskan
perusahaan dari pajak penghasilan untuk periode tertentu,
sering diberikan untuk mendorong investasi.")
print(f" - Dengan tidak adanya beban pajak di 2024 dan 2025,
*laba bersih* melonjak dari Rp780.000.000 (2023) ke
Rp1.728.125.000 (2025), hampir setara dengan *laba
operasional*, menunjukkan dampak besar dari kebijakan ini.")
print("Secara keseluruhan, grafik ini menyoroti tren
pertumbuhan keuangan dan dampak signifikan *tax holiday* pada
profitabilitas bersih.")

# --- Tampilkan Tabel Data ---
print("\nTabel Data:")
print(df.to_string(index=False))

```

- ***Simulasi Depresiasi***

```

import pandas as pd
import matplotlib.pyplot as plt
import io

# Data dari CSV

```



```

data = """
tahun,transaksi_pendapatan,transaksi_beban_operasional,depresiasi_garis_lurus,sisa_garis_lurus,depresiasi_saldo_menurun,sisa_saldo_menurun,total_depresiasi,kebijakan_fiskal_tax_rate
2023,5500000000,3200000000,550000000.0,8950000000.0,750000000,2250000000,1300000000.0,0.22
2024,6000000000,3500000000,550000000.0,8400000000.0,562500000,1687500000,1112500000.0,0.22
2025,6500000000,3800000000,550000000.0,7850000000.0,421875000,1265625000,971875000.0,0.22
"""

try:
    # Membaca data ke DataFrame
    df = pd.read_csv(io.StringIO(data))

    # Memastikan kolom numerik
    numeric_cols = ['depresiasi_garis_lurus',
'sisa_garis_lurus', 'depresiasi_saldo_menurun',
'sisa_saldo_menurun', 'total_depresiasi']
    for col in numeric_cols:
        df[col] = pd.to_numeric(df[col], errors='coerce')

    # Visualisasi
    plt.figure(figsize=(12, 8))

    tahun_data = df['tahun']
    dep_garis_lurus = df['depresiasi_garis_lurus']
    dep_saldo_menurun = df['depresiasi_saldo_menurun']
    total_dep = df['total_depresiasi']
    sisa_garis_lurus = df['sisa_garis_lurus']
    sisa_saldo_menurun = df['sisa_saldo_menurun']

    plt.plot(tahun_data, dep_garis_lurus, marker='o',
label='Depresiasi Garis Lurus', color='blue')
    for x, y in zip(tahun_data, dep_garis_lurus):
        if pd.notna(y):
            plt.text(x, y + y*0.05, f'{y:,.0f}'.replace(',',','),
'.'), ha='center', va='bottom', fontsize=8)

    plt.plot(tahun_data, dep_saldo_menurun, marker='o',
label='Depresiasi Saldo Menurun', color='red')
    for x, y in zip(tahun_data, dep_saldo_menurun):
        if pd.notna(y):
            plt.text(x, y + y*0.05, f'{y:,.0f}'.replace(',',','),
'.'), ha='center', va='bottom', fontsize=8)

```

```

plt.plot(tahun_data, total_dep, marker='o', label='Total
Depresiasi', color='green', linewidth=2)
for x, y in zip(tahun_data, total_dep):
    if pd.notna(y):
        plt.text(x, y + y*0.03, f'{y:,.0f}'.replace(',',
'.'), ha='center', va='bottom', fontsize=9, fontweight='bold')

plt.title('Tren Depresiasi Aset (2023-2025)', fontsize=14)
plt.xlabel('Tahun', fontsize=12)
plt.ylabel('Depresiasi (Rupiah)', fontsize=12)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

except Exception as e:
    print(f"Terjadi error: {e}")
    print("Mohon periksa data atau lingkungan Anda.")

```

- Visualisasi 3D Depresiasi

```

import pandas as pd
import plotly.graph_objects as go
import io

# Data dari CSV yang diberikan
data = """
tahun,transaksi_pendapatan,transaksi_beban_operasional,de
presiasi_garis_lurus,sisa_garis_lurus,depresiasi_saldo_me
nurun,sisa_saldo_menurun,total_depresiasi,kebijakan_fiska
l_tax_rate
2023,5500000000,3200000000,550000000.0,8950000000.0,75000
0000,2250000000,1300000000.0,0.22
2024,6000000000,3500000000,550000000.0,8400000000.0,56250
0000,1687500000,1112500000.0,0.22
2025,6500000000,3800000000,550000000.0,7850000000.0,42187
5000,1265625000,971875000.0,0.22
"""

try:
    # Membaca data ke DataFrame
    df = pd.read_csv(io.StringIO(data))

    # Memastikan kolom numerik
    numeric_cols = [
        'depresiasi_garis_lurus',
        'depresiasi_saldo_menurun',
        'total_depresiasi'
    ]

```

```

]
for col in numeric_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Data untuk plot
years = df['tahun'].values
metrics_to_plot = ['depresiasi_garis_lurus',
'depresiasi_saldo_menurun', 'total_depresiasi']
metric_labels = ['Depresiasi Garis Lurus',
'Depresiasi Saldo Menurun', 'Total Depresiasi']

# Warna untuk setiap garis
colors = ['#1f77b4', '#ff7f0e', '#2ca02c']

data_plot = []

for i, metric in enumerate(metrics_to_plot):
    data_plot.append(go.Scatter3d(
        x=years,
        y=[i] * len(years),
        z=df[metric].values,
        mode='markers+lines',
        name=metric_labels[i],
        line=dict(
            color=colors[i],
            width=3
        ),
        marker=dict(
            size=8,
            color=colors[i],
            opacity=0.9,
            symbol='circle'
        ),
        text=[f'{metric_labels[i]} - {year}:
Rp{value:,.0f}'.replace(",",".")
            for year, value in zip(years,
df[metric].values)],
        hoverinfo='text'
    ))

# Layout untuk plot
layout = go.Layout(
    margin=dict(l=0, r=0, b=0, t=60),
    scene=dict(
        xaxis=dict(
            tickvals=years,
            ticktext=years,
            title='Tahun',

```

```

        backgroundcolor="#f0f0f0",
        gridcolor="#d3d3d3",
        showbackground=True,
        zerolinecolor="#d3d3d3"
    ),
    yaxis=dict(
        tickvals=list(range(len(metrics_to_plot)))
    ),

    ticktext=metric_labels,
    title='Metode Depresiasi',
    backgroundcolor="#fafafa",
    gridcolor="#d3d3d3",
    showbackground=True,
    zerolinecolor="#d3d3d3"
    ),
    zaxis=dict(
        title='Jumlah Depresiasi (Rupiah)',
        tickformat=',',
        tickprefix='Rp',
        backgroundcolor="#e0e0e0",
        gridcolor="#d3d3d3",
        showbackground=True,
        zerolinecolor="#d3d3d3"
    ),
    camera=dict(
        up=dict(x=0, y=0, z=1),
        center=dict(x=0, y=0, z=0),
        eye=dict(x=1.5, y=1.5, z=0.8)
    )
),
title=dict(
    text='Perbandingan Metode Depresiasi Tahunan
3D (Interaktif)',
    y=0.95,
    x=0.5,
    xanchor='center',
    yanchor='top',
    font=dict(
        size=20,
        color='#333'
    )
),
hovermode='closest',
showlegend=True
)

fig = go.Figure(data=data_plot, layout=layout)
fig.show()

```

```

## Deskripsi Grafik 3D Interaktif Depresiasi
print("\n## Deskripsi Grafik 3D Interaktif
Depresiasi")

print("\n**Deskripsi Singkat:**")
print("Grafik 3D interaktif ini memvisualisasikan
besaran biaya depresiasi per tahun dari 2023 hingga 2025,
membandingkan antara metode depresiasi garis lurus, saldo
menurun, dan total depresiasi.")
print("Sumbu X merepresentasikan Tahun (2023, 2024,
2025).")
print("Sumbu Y merepresentasikan Metode Depresiasi
(Garis Lurus, Saldo Menurun, Total).")
print("Sumbu Z merepresentasikan Jumlah Depresiasi
dalam Rupiah.")

print("\n**Maksud dan Penjelasan Grafik:**")
print("Grafik interaktif ini memberikan cara yang
lebih dinamis untuk memahami bagaimana jumlah depresiasi
berfluktuasi dari tahun ke tahun dan bagaimana
perbandingannya antar metode.")
print("- Setiap garis dan titik berwarna
merepresentasikan biaya depresiasi untuk metode tertentu
selama periode tiga tahun.")
print("- **Garis Biru (Garis Lurus):** Menunjukkan
tren depresiasi yang konstan di Rp550.000.000 setiap
tahun.")
print("- **Garis Oranye (Saldo Menurun):**
Menunjukkan tren depresiasi yang lebih tinggi di awal
(Rp750.000.000 di 2023) dan menurun (Rp562.500.000 di
2024, Rp421.875.000 di 2025).")
print("- **Garis Hijau (Total):** Menunjukkan total
biaya depresiasi yang menurun dari Rp1.300.000.000 di
2023 menjadi Rp971.875.000 di 2025.")

print("\nBerdasarkan data pada grafik:")
print("- Depresiasi garis lurus tetap stabil di
Rp550.000.000 setiap tahun, menunjukkan konsistensi
metode ini.")
print("- Depresiasi saldo menurun menurun secara
konsisten, sesuai dengan karakteristik metode tersebut.")
print("- Total depresiasi menurun dari tahun ke
tahun, dipengaruhi oleh penurunan depresiasi saldo
menurun.")
print("- Arahkan kursor ke setiap titik data untuk
melihat nilai pastinya.")

```

```
except Exception as e:
    print(f"Terjadi error: {e}")
    print("Mohon periksa kembali data atau lingkungan
Anda.")
```

- ***Simulasi Tax Holiday***

```
# Import libraries
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Data dari hasil query
data = {
    'Tahun': ['2023', '2024', '2025'],
    'Pajak_Tanpa_Tax_Holiday': [220000000, 305250000,
380187500],
    'Pajak_Dengan_Tax_Holiday': [220000000, 0, 0],
    'Laba_Bersih_Tanpa_Tax_Holiday': [780000000, 1082250000,
1347937500],
    'Laba_Bersih_Dengan_Tax_Holiday': [780000000, 1387500000,
1728125000]
}

df = pd.DataFrame(data)

# Set style and font (DejaVu Sans is default, widely supported)
sns.set_style("whitegrid")
plt.rcParams['font.family'] = 'DejaVu Sans'
plt.rcParams['font.size'] = 12

# Plot 1: Bar Chart for Tax Comparison
plt.figure(figsize=(10, 6))
bar_width = 0.35
index = range(len(df['Tahun']))

plt.bar(index, df['Pajak_Tanpa_Tax_Holiday'], bar_width,
label='Tanpa Tax Holiday', color='#ff9999', edgecolor='white')
plt.bar([i + bar_width for i in index],
df['Pajak_Dengan_Tax_Holiday'], bar_width, label='Dengan Tax
Holiday', color='#66b3ff', edgecolor='white')

plt.title('Perbandingan Pajak dengan dan tanpa Tax Holiday
(2023-2025)', fontsize=16, weight='bold', pad=20)
plt.xlabel('Tahun', fontsize=14)
plt.ylabel('Pajak (Rp)', fontsize=14)
plt.xticks([i + bar_width/2 for i in index], df['Tahun'])
plt.legend(title='Skenario', loc='upper right', fontsize=12)
```

```

plt.ticklabel_format(style='plain', axis='y')
plt.gca().yaxis.set_major_formatter(plt.FuncFormatter(lambda x,
_: f'{int(x/1000000)}M'))

# Add value labels on bars
for i in index:
    plt.text(i, df['Pajak_Tanpa_Tax_Holiday'][i] + 10000000,
f'{int(df["Pajak_Tanpa_Tax_Holiday"][i]/1000000)}M',
ha='center', fontsize=10)
    plt.text(i + bar_width, df['Pajak_Dengan_Tax_Holiday'][i] +
10000000, f'{int(df["Pajak_Dengan_Tax_Holiday"][i]/1000000)}M',
ha='center', fontsize=10)

plt.tight_layout()
plt.show()

# Plot 2: Line Chart for Net Profit Trend
plt.figure(figsize=(10, 6))
plt.plot(df['Tahun'], df['Laba_Bersih_Tanpa_Tax_Holiday'],
marker='o', label='Tanpa Tax Holiday', color='#ff9999',
linewidth=2.5, markersize=8)
plt.plot(df['Tahun'], df['Laba_Bersih_Dengan_Tax_Holiday'],
marker='o', label='Dengan Tax Holiday', color='#66b3ff',
linewidth=2.5, markersize=8)

plt.title('Tren Laba Bersih dengan dan tanpa Tax Holiday (2023-
2025)', fontsize=16, weight='bold', pad=20)
plt.xlabel('Tahun', fontsize=14)
plt.ylabel('Laba Bersih (Rp)', fontsize=14)
plt.legend(title='Skenario', loc='lower right', fontsize=12)
plt.ticklabel_format(style='plain', axis='y')
plt.gca().yaxis.set_major_formatter(plt.FuncFormatter(lambda x,
_: f'{int(x/1000000)}M'))

# Add value labels
for i, txt in enumerate(df['Laba_Bersih_Tanpa_Tax_Holiday']):
    plt.text(df['Tahun'][i], txt + 50000000,
f'{int(txt/1000000)}M', ha='center', fontsize=10)
for i, txt in enumerate(df['Laba_Bersih_Dengan_Tax_Holiday']):
    plt.text(df['Tahun'][i], txt + 50000000,
f'{int(txt/1000000)}M', ha='center', fontsize=10)

plt.tight_layout()
plt.show()

```

- Visualisasi 3D Tax Holiday

```
import pandas as pd
```

```

import plotly.graph_objects as go
import io
import numpy as np

# Data dari hasil perhitungan
data = {
    'Tahun': ['2023', '2024', '2025'],
    'Pajak_Tanpa_Tax_Holiday': [220000000, 305250000,
380187500],
    'Pajak_Dengan_Tax_Holiday': [220000000, 0, 0],
    'Laba_Bersih_Tanpa_Tax_Holiday': [780000000,
1082250000, 1347937500],
    'Laba_Bersih_Dengan_Tax_Holiday': [780000000,
1387500000, 1728125000]
}

df = pd.DataFrame(data)

# Define the years and scenarios
years = df['Tahun'].tolist()
tax_scenarios = ['Tanpa Tax Holiday', 'Dengan Tax
Holiday']
laba_scenarios = ['Tanpa Tax Holiday', 'Dengan Tax
Holiday']

# Function to create a cuboid (bar) for Mesh3d
def create_cuboid(x_center, y_center, z_height, width,
depth, color, name, hovertext):
    x = [x_center - width/2, x_center + width/2, x_center
+ width/2, x_center - width/2,
        x_center - width/2, x_center + width/2, x_center
+ width/2, x_center - width/2]
    y = [y_center - depth/2, y_center - depth/2, y_center
+ depth/2, y_center + depth/2,
        y_center - depth/2, y_center - depth/2, y_center
+ depth/2, y_center + depth/2]
    z = [0, 0, 0, 0, z_height, z_height, z_height,
z_height]

    i = [7, 0, 0, 0, 4, 4, 6, 6, 4, 0, 3, 2]
    j = [3, 4, 1, 2, 5, 6, 5, 2, 0, 1, 6, 3]
    k = [0, 7, 2, 3, 6, 7, 1, 1, 5, 5, 7, 6]

    return go.Mesh3d(x=x, y=y, z=z, i=i, j=j, k=k,
color=color, name=name, hoverinfo='text', text=hovertext,
opacity=0.9)

```



```

# --- 3D Interactive Bar Chart for Tax Comparison using
Mesh3d ---
tax_data_plot_3d = []
bar_width = 0.8
bar_depth = 0.4

year_map = {year: i for i, year in enumerate(years)}
tax_scenario_map = {scenario: i for i, scenario in
enumerate(tax_scenarios)}

for year_str in years:
    year_pos = year_map[year_str]
    # Tax Without Tax Holiday
    value_tanpa = df[df['Tahun'] ==
year_str]['Pajak_Tanpa_Tax_Holiday'].iloc[0]
    scenario_pos_tanpa = tax_scenario_map['Tanpa Tax
Holiday']
    hover_tanpa = f"Tahun: {year_str}<br>Skenario: Tanpa
Tax Holiday<br>Pajak: Rp{value_tanpa:,.0f}".replace(',', ' ',
'.')
    tax_data_plot_3d.append(create_cuboid(year_pos,
scenario_pos_tanpa, value_tanpa, bar_width, bar_depth,
'#ff9999', f'{year_str} - Tanpa TH', hover_tanpa))

    # Tax With Tax Holiday
    value_dengan = df[df['Tahun'] ==
year_str]['Pajak_Dengan_Tax_Holiday'].iloc[0]
    scenario_pos_dengan = tax_scenario_map['Dengan Tax
Holiday']
    hover_dengan = f"Tahun: {year_str}<br>Skenario:
Dengan Tax Holiday<br>Pajak:
Rp{value_dengan:,.0f}".replace(',', ' ', '.')
    tax_data_plot_3d.append(create_cuboid(year_pos,
scenario_pos_dengan, value_dengan, bar_width, bar_depth,
'#66b3ff', f'{year_str} - Dengan TH', hover_dengan))

fig_tax_3d_mesh = go.Figure(data=tax_data_plot_3d,
layout=go.Layout(
    title='Perbandingan Pajak dengan dan tanpa Tax
Holiday (3D Interaktif)',
    scene=dict(
        xaxis=dict(title='Tahun',
tickvals=list(year_map.values()), ticktext=years),
        yaxis=dict(title='Skenario Pajak',
tickvals=list(tax_scenario_map.values()),
ticktext=tax_scenarios),
        zaxis=dict(title='Pajak (Rupiah)',
tickformat=', '),

```

```

        camera=dict(
            up=dict(x=0, y=0, z=1),
            center=dict(x=0, y=0, z=0),
            eye=dict(x=1.8, y=1.8, z=0.8)
        )
    ),
    margin=dict(l=0, r=0, b=0, t=40)
))

# Show the 3D tax plot
print("\n## Grafik 3D Interaktif Perbandingan Pajak
(Mesh3d)")
try:
    fig_tax_3d_mesh.show()
except Exception as e:
    print(f"Error showing 3D Tax plot: {e}")

# --- 3D Interactive Bar Chart for Net Profit Comparison
using Mesh3d ---
laba_data_plot_3d = []
laba_scenario_map = {scenario: i for i, scenario in
enumerate(laba_scenarios)}

for year_str in years:
    year_pos = year_map[year_str]
    # Laba Bersih Tanpa Tax Holiday
    value_tanpa = df[df['Tahun'] ==
year_str]['Laba Bersih Tanpa Tax Holiday'].iloc[0]
    scenario_pos_tanpa = laba_scenario_map['Tanpa Tax
Holiday']
    hover_tanpa = f"Tahun: {year_str}<br>Skenario: Tanpa
Tax Holiday<br>Laba Bersih:
Rp{value_tanpa:,.0f}".replace(',', ', ')
    laba_data_plot_3d.append(create_cuboid(year_pos,
scenario_pos_tanpa, value_tanpa, bar_width, bar_depth,
'#ff9999', f'{year_str} - Tanpa TH', hover_tanpa))

    # Laba Bersih Dengan Tax Holiday
    value_dengan = df[df['Tahun'] ==
year_str]['Laba Bersih Dengan Tax Holiday'].iloc[0]
    scenario_pos_dengan = laba_scenario_map['Dengan Tax
Holiday']
    hover_dengan = f"Tahun: {year_str}<br>Skenario:
Dengan Tax Holiday<br>Laba Bersih:
Rp{value_dengan:,.0f}".replace(',', ', ')
    laba_data_plot_3d.append(create_cuboid(year_pos,
scenario_pos_dengan, value_dengan, bar_width, bar_depth,
'#66b3ff', f'{year_str} - Dengan TH', hover_dengan))

```

```

fig_laba_3d_mesh = go.Figure(data=laba_data_plot_3d,
layout=go.Layout(
    title='Tren Laba Bersih dengan dan tanpa Tax Holiday
(3D Interaktif)',
    scene=dict(
        xaxis=dict(title='Tahun',
tickvals=list(year_map.values()), ticktext=years),
        yaxis=dict(title='Skenario Laba Bersih',
tickvals=list(laba_scenario_map.values()),
ticktext=laba_scenarios),
        zaxis=dict(title='Laba Bersih (Rupiah)',
tickformat=', '),
        camera=dict(
            up=dict(x=0, y=0, z=1),
            center=dict(x=0, y=0, z=0),
            eye=dict(x=1.8, y=1.8, z=0.8)
        )
    ),
    margin=dict(l=0, r=0, b=0, t=40)
))

# Show the 3D net profit plot
print("\n## Grafik 3D Interaktif Tren Laba Bersih
(Mesh3d)")
try:
    fig_laba_3d_mesh.show()
except Exception as e:
    print(f"Error showing 3D Net Profit plot: {e}")

# --- Deskripsi untuk Grafik 3D Interaktif ---
print("\n## Deskripsi dan Penjelasan Grafik 3D Interaktif
(Mesh3d)")

print("\nGrafik-grafik di atas disajikan dalam format 3D
interaktif menggunakan Plotly:")

print("\n**Cara Berinteraksi:**")
print("- **Memutar/Melihat dari Sudut Berbeda:** Klik dan
seret (drag) pada area grafik.")
print("- **Memperbesar/Memperkecil (Zoom):** Gunakan roda
gulir (scroll) pada mouse atau gerakan mencubit (pinch)
pada trackpad.")
print("- **Menggeser (Pan):** Klik tombol tengah mouse
(atau klik kanan dan seret) atau gunakan kombinasi tombol
(misalnya Shift + klik kiri dan seret).")

```

```

print("- **Informasi Detail (Hover):** Arahkan kursor
(mouse) ke atas 'batang' untuk melihat nilai pastinya
dalam format Rupiah.")
print("- **Tombol Mode:** Di pojok kanan atas grafik, ada
ikon-ikon kecil (modebar Plotly) yang memungkinkan Anda
melakukan berbagai tindakan seperti menyimpan gambar,
reset tampilan, zoom, pan, dll.")

print("\n**Penjelasan Masing-masing Grafik:**")

print("\n### Grafik 3D Perbandingan Pajak:")
print("- Grafik ini menggunakan bentuk 3D (Mesh3d)
menyerupai batang untuk membandingkan jumlah pajak per
tahun (Sumbu X, direpresentasikan secara numerik) antara
skenario 'Tanpa Tax Holiday' dan 'Dengan Tax Holiday'
(Sumbu Y, direpresentasikan secara numerik). Label sumbu
disesuaikan agar mudah dibaca.")
print("- Tinggi batang (Sumbu Z) merepresentasikan
besaran pajak dalam Rupiah.")
print("- **Warna Merah Muda:** Menunjukkan skenario
'Tanpa Tax Holiday', dengan nilai pajak meningkat dari
Rp220.000.000 di 2023 menjadi Rp380.187.500 di 2025.")
print("- **Warna Biru:** Menunjukkan skenario 'Dengan Tax
Holiday', dengan pajak Rp220.000.000 di 2023, dan nol
untuk 2024 dan 2025.")
print("- Grafik ini secara dramatis menunjukkan bagaimana
*tax holiday* menghilangkan beban pajak di tahun 2024 dan
2025.")

print("\n### Grafik 3D Tren Laba Bersih:")
print("- Grafik ini juga menggunakan bentuk 3D menyerupai
batang untuk menunjukkan tren Laba Bersih per tahun
(Sumbu X) untuk kedua skenario (Sumbu Y).")
print("- Tinggi batang (Sumbu Z) merepresentasikan jumlah
laba bersih dalam Rupiah.")
print("- **Warna Merah Muda:** Menunjukkan laba bersih
pada skenario 'Tanpa Tax Holiday', meningkat dari
Rp780.000.000 di 2023 menjadi Rp1.347.937.500 di 2025.")
print("- **Warna Biru:** Menunjukkan laba bersih pada
skenario 'Dengan Tax Holiday', dengan lonjakan signifikan
di 2024 (Rp1.387.500.000) dan 2025 (Rp1.728.125.000)
karena pajak nol.")
print("- Grafik ini dengan jelas menggambarkan lonjakan
laba bersih di tahun 2024 dan 2025 pada skenario 'Dengan
Tax Holiday' sebagai akibat langsung dari beban pajak
yang menjadi nol.")

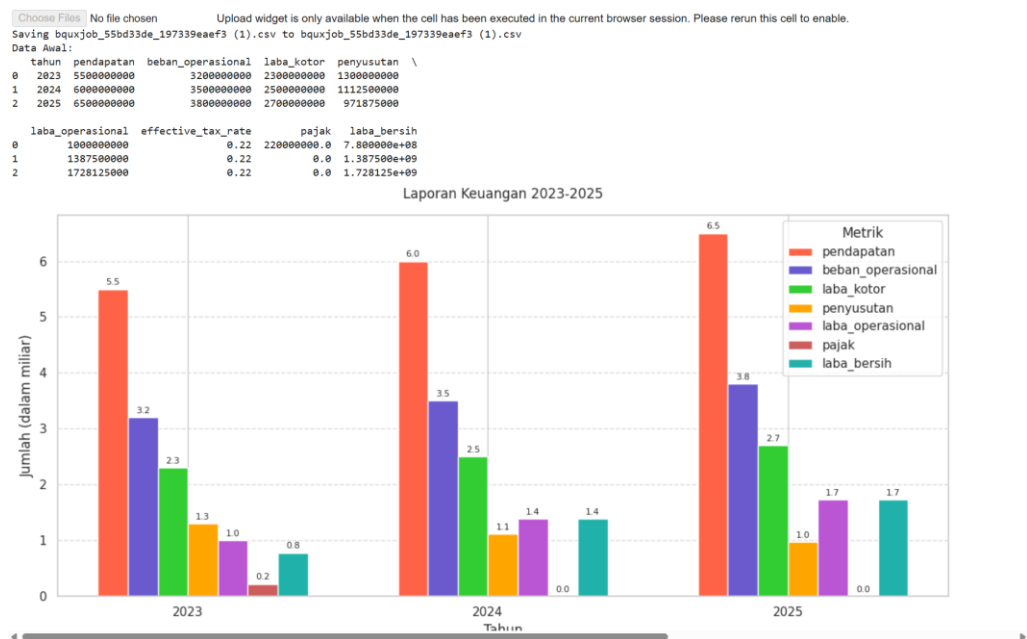
```

```
print("\nSecara keseluruhan, penggunaan grafik 3D
interaktif dengan Mesh3d ini memungkinkan representasi
data finansial sebagai 'batang' solid di ruang 3D,
membuatnya lebih visual dan mudah dipahami, serta
mempertahankan interaktivitas Plotly.")
```

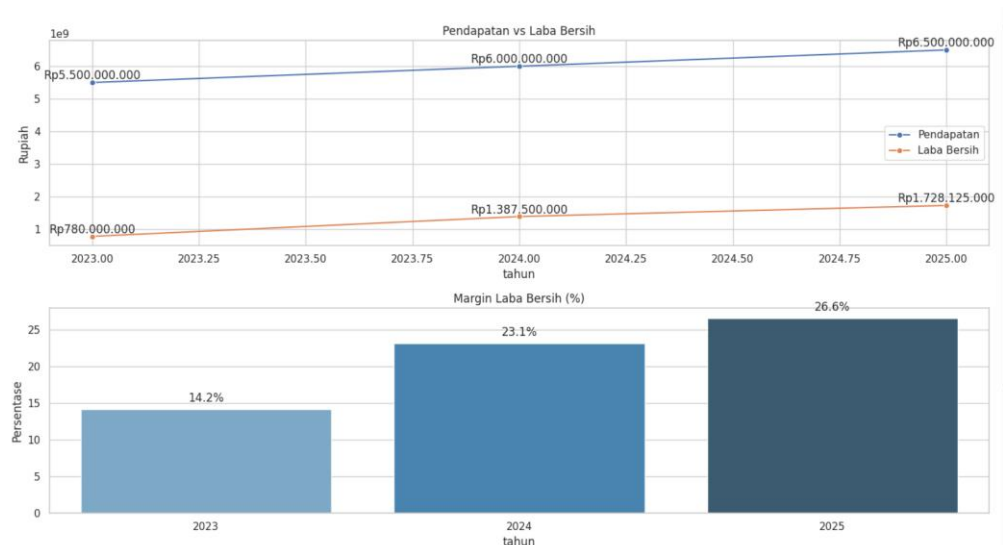
5. Memvisualisasikan Data dengan Lebih Fleksibel

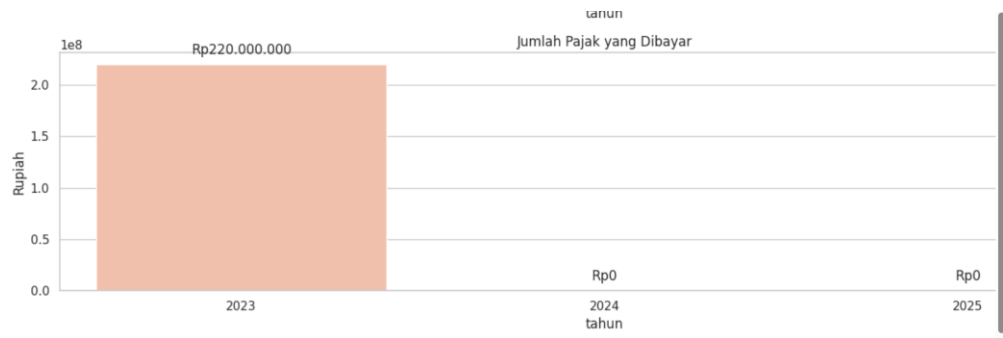
Dengan pustaka visualisasi seperti matplotlib dan seaborn, Google Colab memungkinkan pembuatan grafik yang disesuaikan, seperti tren laba bersih tahunan, perbandingan PPh Badan antar-skenario (normal, tax holiday, atau metode depresiasi berbeda).

Simulasi L/R



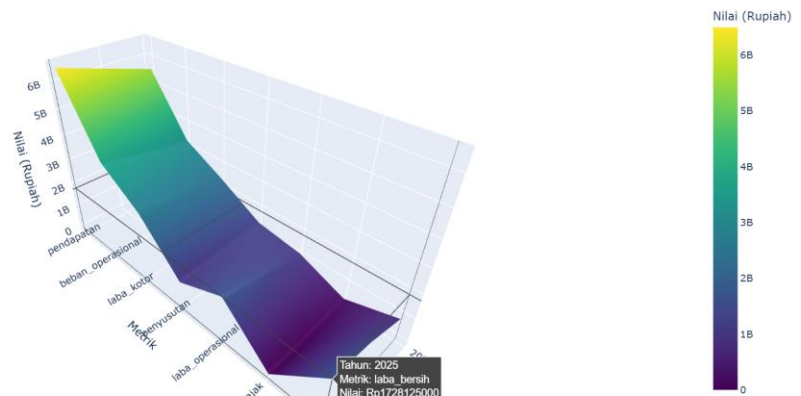
Data Analitik Sederhana untuk Simulasi Laba Rugi





Visualisasi 3D dan Penjelasan Singkat Grafik

Analisis Keuangan Tahunan 3D Interaktif (Permukaan)



Deskripsi dan Penjelasan Grafik 3D (Plotly)

Deskripsi Grafik:

Grafik ini menampilkan laporan keuangan tahunan (2023-2025) dalam format 3D interaktif.

Permukaan grafik merepresentasikan nilai metrik keuangan tertentu (ditunjukkan pada Sumbu Y) untuk setiap tahun (ditunjukkan pada Sumbu X), dengan ketinggian permukaan Anda dapat memutar grafik dengan mengklik dan menyeret untuk melihat data dari berbagai sudut.

Hover (arahkan kursor) ke permukaan untuk melihat detail angka lengkapnya dalam format Rupiah.

Interpretasi Data:

- Grafik menunjukkan kenaikan progresif pada "pendapatan" dari Rp5.500.000.000 (2023) ke Rp6.500.000.000 (2025), serta "beban operasional" dari Rp3.200.000.000 ke Rp3.800.000.000.
- Peningkatan ini menghasilkan kenaikan stabil pada "laba kotor" dari Rp2.300.000.000 ke Rp2.700.000.000.
- "Penyusutan" menurun dari Rp1.300.000.000 (2023) ke Rp971.875.000 (2025), sesuai dengan metode saldo menurun ganda (DDG), mengindikasikan pengurangan nilai aset sej
- "Laba operasional" meningkat signifikan dari Rp1.000.000.000 (2023) ke Rp1.728.125.000 (2025), didorong oleh penurunan penyusutan.

Deskripsi Grafik:

- Grafik ini menampilkan laporan keuangan tahunan (2023-2025) dalam format 3D interaktif.
- Permukaan grafik merepresentasikan nilai metrik keuangan tertentu (ditunjukkan pada Sumbu Y) untuk setiap tahun (ditunjukkan pada Sumbu X), dengan ketinggian permukaan menunjukkan jumlah nilai dalam Rupiah (ditunjukkan pada Sumbu Z).

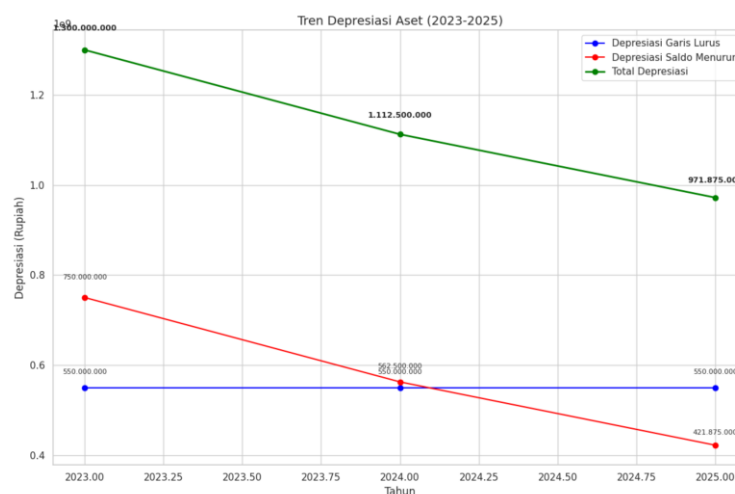
Interpretasi Data:

- Grafik menunjukkan kenaikan progresif pada pendapatan dari Rp5.500.000.000 (2023) ke Rp6.500.000.000 (2025), serta beban operasional dari Rp3.200.000.000 ke Rp3.800.000.000, mencerminkan pertumbuhan bisnis.

- Peningkatan ini menghasilkan kenaikan stabil pada laba kotor dari Rp2.300.000.000 ke Rp2.700.000.000.
- Penyusutan menurun dari Rp1.300.000.000 (2023) ke Rp971.875.000 (2025), sesuai dengan metode saldo menurun ganda (DDB), mengindikasikan pengurangan nilai aset seiring waktu.
- Laba operasional meningkat signifikan dari Rp1.000.000.000 (2023) ke Rp1.728.125.000 (2025), didorong oleh penurunan penyusutan.
- Aspek mencolok adalah nilai pajak yang nol untuk tahun 2024 dan 2025 (dibandingkan Rp220.000.000 pada 2023), kemungkinan akibat *tax holiday*.
- Tax holiday adalah fasilitas yang membebaskan perusahaan dari pajak penghasilan untuk periode tertentu, sering diberikan untuk mendorong investasi.
- Dengan tidak adanya beban pajak di 2024 dan 2025, laba bersih melonjak dari Rp780.000.000 (2023) ke Rp1.728.125.000 (2025), hampir setara dengan laba operasional, menunjukkan dampak besar dari kebijakan ini.
- Secara keseluruhan, grafik ini menyoroti tren pertumbuhan keuangan dan dampak signifikan tax holiday pada profitabilitas bersih.

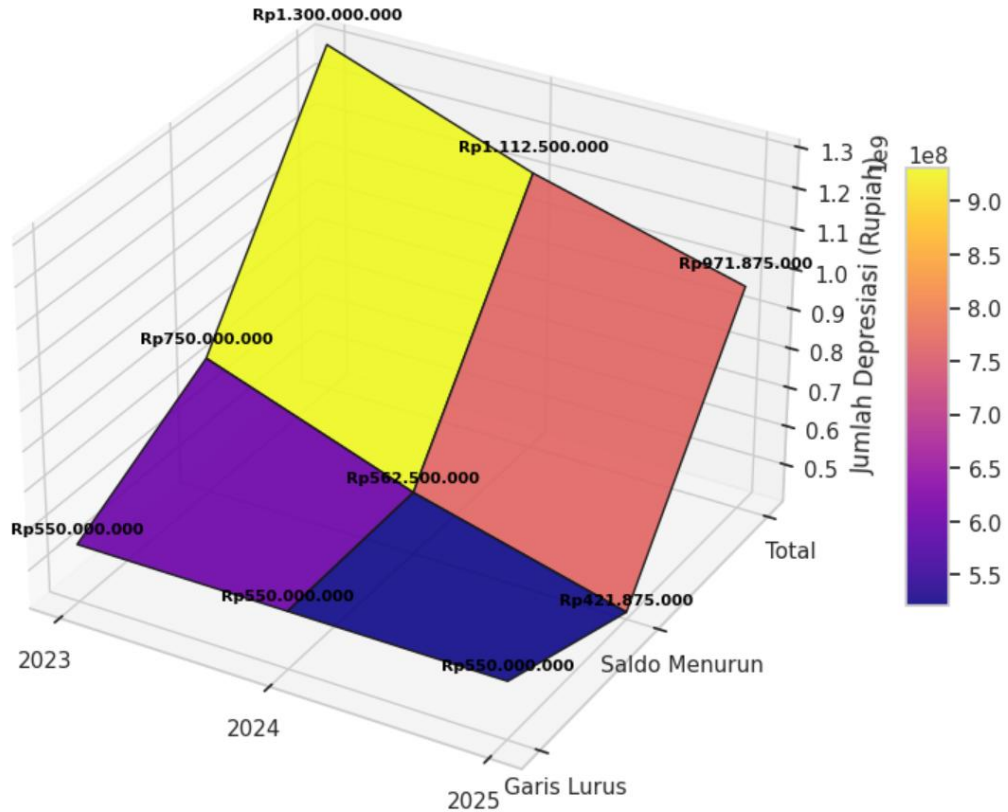
Simulasi Depresiasi

Data Analitik Sederhana untuk Simulasi Depresiasi



Visualisasi 3D dan Deskripsi Singkat

Perbandingan Metode Depresiasi Tahunan 3D



Deskripsi Grafik 3D Depresiasi

Deskripsi Singkat:

Grafik 3D ini memvisualisasikan besaran biaya depresiasi per tahun dari 2023 hingga 2025, membandingkan antara metode depresiasi garis lurus, saldo menurun, dan total. Sumbu X merepresentasikan Tahun (2023, 2024, 2025). Sumbu Y merepresentasikan Metode Depresiasi (Garis Lurus, Saldo Menurun, Total). Sumbu Z merepresentasikan Jumlah Depresiasi dalam Rupiah. Angka persis nilai depresiasi ditampilkan di atas setiap titik data pada permukaan dalam format Rupiah.

Maksud dan Penjelasan Grafik:

Grafik ini bertujuan untuk memberikan gambaran tiga dimensi tentang bagaimana jumlah depresiasi berfluktuasi dari tahun ke tahun dan bagaimana perbandingannya antara: - **Garis Lurus (bagian bawah permukaan di Sumbu Y):** Menunjukkan depresiasi tetap sebesar Rp550.000.000 per tahun, mencerminkan alokasi biaya yang konstan. - **Saldo Menurun (bagian tengah permukaan di Sumbu Y):** Menunjukkan tren depresiasi yang menurun dari Rp750.000.000 (2023) ke Rp421.875.000 (2025), sesuai dengan - **Total Depresiasi (bagian atas permukaan di Sumbu Y):** Menunjukkan agregat biaya depresiasi yang menurun dari Rp1.300.000.000 (2023) ke Rp971.875.000 (2025), se

Grafik 3D ini membantu dalam:

1. **Memahami Dampak Metode Depresiasi:** Melihat perbedaan pola alokasi biaya antara garis lurus (stabil) dan saldo menurun (menurun).
2. **Menganalisis Tren Total Depresiasi:** Mengamati penurunan total depresiasi yang mencerminkan pengurangan beban depresiasi seiring waktu.
3. **Identifikasi Anomali/Perubahan:** Tidak ada lonjakan signifikan, tetapi penurunan total depresiasi menunjukkan efektivitas metode saldo menurun.

Berdasarkan data pada grafik:

- Depresiasi garis lurus tetap konstan di Rp550.000.000 per tahun, menunjukkan tidak ada penambahan aset baru yang signifikan.
- Depresiasi saldo menurun menurun secara bertahap, konsisten dengan metode DDB untuk aset yang sudah ada.
- Total depresiasi menurun dari Rp1.300.000.000 ke Rp971.875.000, didorong oleh penurunan depresiasi saldo menurun yang lebih besar daripada depresiasi garis lurus.

Deskripsi Singkat:

- Grafik 3D ini memvisualisasikan besaran biaya depresiasi per tahun dari 2023 hingga 2025, membandingkan antara metode depresiasi garis lurus, saldo menurun, dan total depresiasi.
- Sumbu X merepresentasikan Tahun (2023, 2024, 2025).

- Sumbu Y merepresentasikan Metode Depresiasi (Garis Lurus, Saldo Menurun, Total).
- Sumbu Z merepresentasikan Jumlah Depresiasi dalam Rupiah.

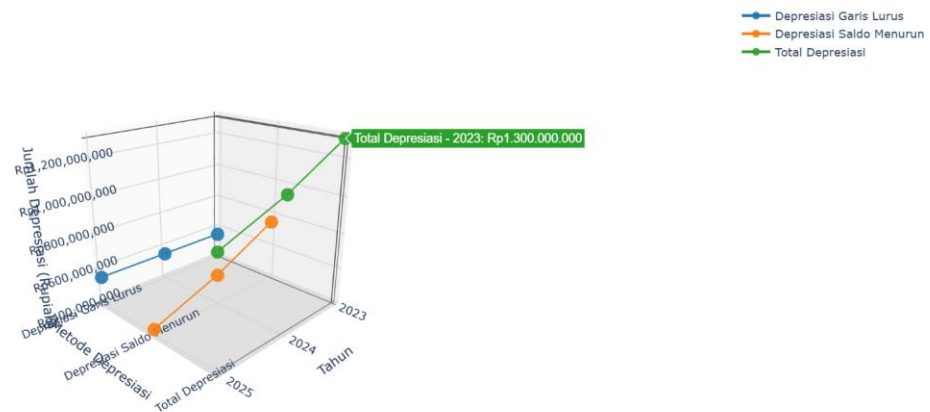
Maksud dan Penjelasan Grafik:

- Grafik ini bertujuan untuk memberikan gambaran tiga dimensi tentang bagaimana jumlah depresiasi berfluktuasi dari tahun ke tahun dan bagaimana perbandingannya antar metode.
- Garis Lurus (bagian bawah permukaan di Sumbu Y): Menunjukkan depresiasi tetap sebesar Rp550.000.000 per tahun, mencerminkan alokasi biaya yang konstan.
- Saldo Menurun (bagian tengah permukaan di Sumbu Y): Menunjukkan tren depresiasi yang menurun dari Rp750.000.000 (2023) ke Rp421.875.000 (2025), sesuai dengan metode saldo menurun ganda.
- Total Depresiasi (bagian atas permukaan di Sumbu Y): Menunjukkan agregat biaya depresiasi yang menurun dari Rp1.300.000.000 (2023) ke Rp971.875.000 (2025), sebagai kombinasi metode garis lurus dan saldo menurun.

Berdasarkan data pada grafik:

- Depresiasi garis lurus tetap konstan di Rp550.000.000 per tahun, menunjukkan tidak ada penambahan aset baru yang signifikan.
- Depresiasi saldo menurun menurun secara bertahap, konsisten dengan metode DDB untuk aset yang sudah ada.
- Total depresiasi menurun dari Rp1.300.000.000 ke Rp971.875.000, didorong oleh penurunan depresiasi saldo menurun yang lebih besar daripada depresiasi garis lurus yang stabil.

Perbandingan Metode Depresiasi Tahunan 3D (Interaktif)



Deskripsi Grafik 3D Interaktif Depresiasi

Deskripsi Singkat:

Grafik 3D interaktif ini memvisualisasikan besaran biaya depresiasi per tahun dari 2023 hingga 2025, membandingkan antara metode depresiasi garis lurus, saldo menurun. Sumbu X merepresentasikan Tahun (2023, 2024, 2025).

Sumbu Y merepresentasikan Metode Depresiasi (Garis Lurus, Saldo Menurun, Total). Sumbu Z merepresentasikan Jumlah Depresiasi dalam Rupiah.

Maksud dan Penjelasan Grafik:

Grafik interaktif ini memberikan cara yang lebih dinamis untuk memahami bagaimana jumlah depresiasi berfluktuasi dari tahun ke tahun dan bagaimana perbandingannya antar metode.

- Setiap garis dan titik berwarna merepresentasikan biaya depresiasi untuk metode tertentu selama periode tiga tahun.

- **Garis Biru (Garis Lurus):** Menunjukkan tren depresiasi yang konstan di Rp550.000.000 setiap tahun.

- **Garis Oranye (Saldo Menurun):** Menunjukkan tren depresiasi yang lebih tinggi di awal (Rp750.000.000 di 2023) dan menurun (Rp562.500.000 di 2024, Rp421.875.000 di 2025).

- **Garis Hijau (Total):** Menunjukkan total biaya depresiasi yang menurun dari Rp1.300.000.000 di 2023 menjadi Rp971.875.000 di 2025.

Maksud dan Penjelasan Grafik:

- Grafik interaktif ini memberikan cara yang lebih dinamis untuk memahami bagaimana jumlah depresiasi berfluktuasi dari tahun ke tahun dan bagaimana perbandingannya antar metode.
- Setiap garis dan titik berwarna merepresentasikan biaya depresiasi untuk metode tertentu selama periode tiga tahun.
- Garis Biru (Garis Lurus): Menunjukkan tren depresiasi yang konstan di Rp550.000.000 setiap tahun.
- Garis Oranye (Saldo Menurun): Menunjukkan tren depresiasi yang lebih tinggi di awal (Rp750.000.000 di 2023) dan menurun (Rp562.500.000 di 2024, Rp421.875.000 di 2025).
- Garis Hijau (Total): Menunjukkan total biaya depresiasi yang menurun dari Rp1.300.000.000 di 2023 menjadi Rp971.875.000 di 2025.

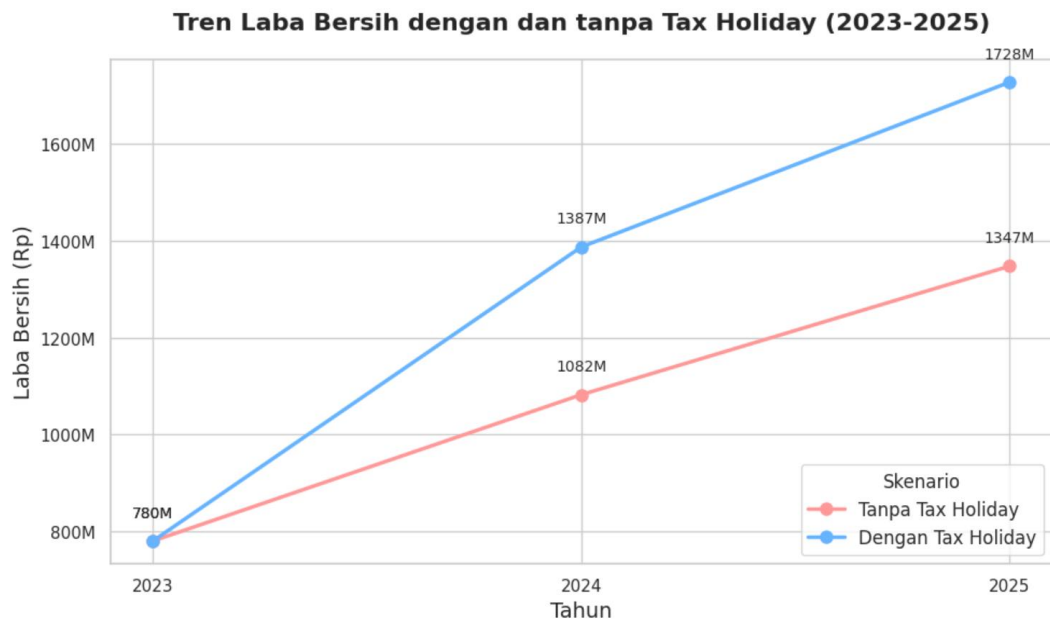
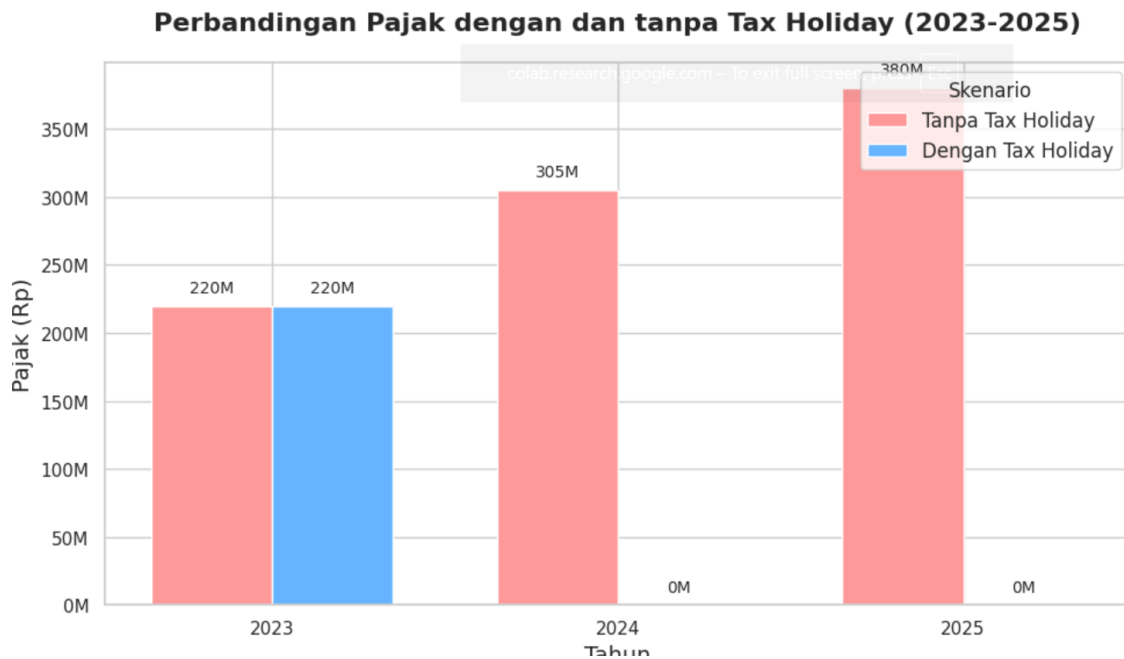
Berdasarkan data pada grafik:

- Depresiasi garis lurus tetap stabil di Rp550.000.000 setiap tahun, menunjukkan konsistensi metode ini.

- Depresiasi saldo menurun menurun secara konsisten, sesuai dengan karakteristik metode tersebut.
- Total depresiasi menurun dari tahun ke tahun, dipengaruhi oleh penurunan depresiasi saldo menurun.

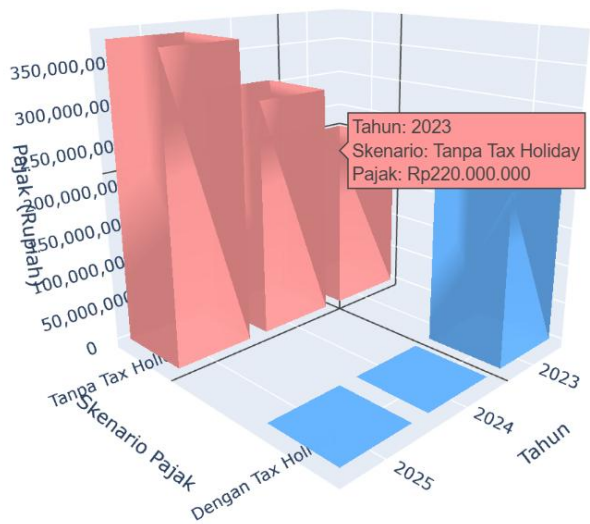
Simulasi Tax Holiday

Data Analitik Sederhana untuk Simulasi Tax Holiday

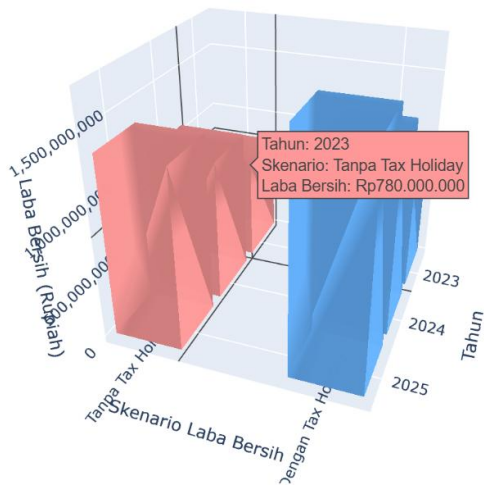


Visualisasi 3D dan Deskripsi Singkat untuk Simulasi Tax Holiday

Perbandingan Pajak dengan dan tanpa Tax Holiday (3D Interaktif)



Tren Laba Bersih dengan dan tanpa Tax Holiday (3D Interaktif)



****Penjelasan Masing-masing Grafik:****

Grafik 3D Perbandingan Pajak:

- Grafik ini menggunakan bentuk 3D (Mesh3d) menyerupai batang untuk membandingkan jumlah pajak per tahun (Sumbu X, direpresentasikan secara numerik)
- Tinggi batang (Sumbu Z) merepresentasikan besaran pajak dalam Rupiah.
- ****Warna Merah Muda:**** Menunjukkan skenario 'Tanpa Tax Holiday', dengan nilai pajak meningkat dari Rp220.000.000 di 2023 menjadi Rp380.187.500 di 2025.
- ****Warna Biru:**** Menunjukkan skenario 'Dengan Tax Holiday', dengan pajak Rp220.000.000 di 2023, dan nol untuk 2024 dan 2025.
- Grafik ini secara dramatis menunjukkan bagaimana *tax holiday* menghilangkan beban pajak di tahun 2024 dan 2025.

Grafik 3D Tren Laba Bersih:

- Grafik ini juga menggunakan bentuk 3D menyerupai batang untuk menunjukkan tren Laba Bersih per tahun (Sumbu X) untuk kedua skenario (Sumbu Y).
- Tinggi batang (Sumbu Z) merepresentasikan jumlah laba bersih dalam Rupiah.
- ****Warna Merah Muda:**** Menunjukkan laba bersih pada skenario 'Tanpa Tax Holiday', meningkat dari Rp780.000.000 di 2023 menjadi Rp1.347.937.500 di 2025.
- ****Warna Biru:**** Menunjukkan laba bersih pada skenario 'Dengan Tax Holiday', dengan lonjakan signifikan di 2024 (Rp1.387.500.000) dan 2025 (Rp1.728.125.000) karena pajak nol.
- Grafik ini dengan jelas menggambarkan lonjakan laba bersih di tahun 2024 dan 2025 pada skenario 'Dengan Tax Holiday' sebagai akibat langsung dari

Secara keseluruhan, penggunaan grafik 3D interaktif dengan Mesh3d ini memungkinkan representasi data finansial sebagai 'batang' solid di ruang 3D,

Penjelasan Masing-masing Grafik:

Grafik 3D Perbandingan Pajak:

- Grafik ini menggunakan bentuk 3D (Mesh3d) menyerupai batang untuk membandingkan jumlah pajak per tahun (Sumbu X, direpresentasikan secara numerik) antara skenario 'Tanpa Tax Holiday' dan 'Dengan Tax Holiday' (Sumbu Y, direpresentasikan secara numerik). Label sumbu disesuaikan agar mudah dibaca.
- Tinggi batang (Sumbu Z) merepresentasikan besaran pajak dalam Rupiah.
- Warna Merah Muda: Menunjukkan skenario 'Tanpa Tax Holiday', dengan nilai pajak meningkat dari Rp220.000.000 di 2023 menjadi Rp380.187.500 di 2025.
- Warna Biru: Menunjukkan skenario 'Dengan Tax Holiday', dengan pajak Rp220.000.000 di 2023, dan nol untuk 2024 dan 2025.
- Grafik ini secara dramatis menunjukkan bagaimana *tax holiday* menghilangkan beban pajak di tahun 2024 dan 2025.

Grafik 3D Tren Laba Bersih:

- Grafik ini juga menggunakan bentuk 3D menyerupai batang untuk menunjukkan tren Laba Bersih per tahun (Sumbu X) untuk kedua skenario (Sumbu Y).
- Tinggi batang (Sumbu Z) merepresentasikan jumlah laba bersih dalam Rupiah.
- Warna Merah Muda: Menunjukkan laba bersih pada skenario 'Tanpa Tax Holiday', meningkat dari Rp780.000.000 di 2023 menjadi Rp1.347.937.500 di 2025.
- Warna Biru: Menunjukkan laba bersih pada skenario 'Dengan Tax Holiday', dengan lonjakan signifikan di 2024 (Rp1.387.500.000) dan 2025 (Rp1.728.125.000) karena pajak nol.
- Grafik ini dengan jelas menggambarkan lonjakan laba bersih di tahun 2024 dan 2025 pada skenario 'Dengan Tax Holiday' sebagai akibat langsung dari beban pajak yang menjadi nol.

C. Kesimpulan

6. Kesimpulan Penggunaan Google Big Query dan Google Colab

Big Query

- Big query akan sangat berguna ketika kita ingin menggunakannya dalam tahap persiapan data dan pemahaman sql, big query ini dapat digunakan untuk menyimpan data dan mengelola tabel seperti tabel transaksi keuangan, aset tetap, dan kebijakan fiskal. BigQuery ideal untuk mendefinisikan struktur dataset dan memastikan data tersedia untuk kueri.
- Menjalankan semua kueri SQL yang disebutkan (SELECT, FROM, WHERE, JOIN, GROUP BY, ORDER BY, CASE) dan juga menjalankan kueri sederhana untuk simulasi seperti laba rugi, depresiasi, dan tax holiday. BigQuery sangat efisien untuk operasi ini, terutama saat menggabungkan tabel atau mengagregasi data besar.
- Sebagai sumber data utama untuk visualisasi di Looker Studio. Looker Studio terintegrasi langsung dengan BigQuery, sehingga bisa digunakan untuk mengambil hasil kueri (laba rugi, depresiasi, serta tax holiday) dan membuat grafik tren atau perbandingan.

Google Colab

- Digunakan untuk visualisasi kustom yang tidak didukung Looker Studio, misalnya grafik kompleks dengan Matplotlib atau Seaborn.
- Analisis lanjutan, misalnya membandingkan skenario secara numerik, membuat visualisasi kustom, atau menulis narasi analitis berdasarkan hasil.