

# **Laporan Analisis Data PPh Badan**

## **Menggunakan Google BigQuery dan Google Colab**

Disusun untuk Memenuhi Tugas Akhir Mata Kuliah Pengkodean dan Pemrograman

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



Oleh:

Nama : Alfina Eka Wardani

NIM : 12030123130120

Kelas : D

**Program Studi Akuntansi**  
**Fakultas Ekonomika dan Bisnis**  
**Universitas Diponegoro**

# **BAB I**

## **Pendahuluan**

### **A. Latar Belakang**

Pemanfaatan teknologi analitik berbasis cloud seperti Google BigQuery dan Google Colab membuka peluang baru dalam dunia pendidikan, khususnya dalam pembelajaran perpajakan yang semakin kompleks. Teknologi ini memungkinkan pengolahan dan analisis data dalam jumlah besar secara cepat dan efisien tanpa memerlukan infrastruktur perangkat keras yang mahal. Melalui Google BigQuery, mahasiswa dapat mengakses, menyimpan, dan menganalisis data keuangan perusahaan dalam skala besar, sementara Google Colab memungkinkan mereka untuk menulis kode dan menjalankan simulasi perpajakan secara interaktif dengan Python, yang kini semakin populer dalam dunia analitik dan akuntansi.

Dalam konteks pembelajaran PPh Badan, penggunaan teknologi ini memberikan pendekatan yang lebih realistis dan aplikatif dibandingkan metode konvensional. Mahasiswa tidak hanya mempelajari teori tentang perhitungan pajak, tetapi juga langsung melakukan simulasi dengan data yang menyerupai kondisi riil perusahaan. Mereka bisa memahami bagaimana setiap kebijakan fiskal dan transaksi keuangan memengaruhi beban pajak suatu entitas. Selain itu, dengan memanfaatkan database dalam Google BigQuery, mahasiswa dapat mencoba berbagai skenario perpajakan dan melakukan analisis perbandingan atas dampaknya terhadap kewajiban PPh Badan secara lebih sistematis.

Lebih jauh lagi, pendekatan berbasis data ini juga menumbuhkan keterampilan analitis dan pemecahan masalah yang sangat dibutuhkan dalam dunia kerja. Mahasiswa diajak untuk berpikir kritis dalam menafsirkan data keuangan dan menentukan perlakuan pajak yang sesuai berdasarkan regulasi yang berlaku. Dengan menggunakan alat seperti Google Colab, mahasiswa dapat belajar menulis skrip yang dapat diubah atau disesuaikan dengan kebutuhan analisis tertentu, memberikan fleksibilitas tinggi dalam proses pembelajaran. Integrasi teknologi ini menjadikan pembelajaran perpajakan lebih dinamis, relevan dengan kebutuhan industri, dan mendorong terciptanya lulusan yang lebih siap menghadapi tantangan profesi di bidang perpajakan dan keuangan.

### **B. Tujuan Pembelajaran**

1. Meningkatkan pemahaman mengenai praktik PPh Badan melalui data.
2. Mengembangkan keterampilan analisis kuantitatif menggunakan SQL dan Python.
3. Menanamkan pemahaman tentang pengaruh kebijakan fiskal melalui skenario simulasi.

## BAB II

### Persiapan Dataset

#### A. Struktur Dataset

##### 1. Tabel Transaksi Keuangan

Kolom: tahun, pendapatan, beban\_operasional, penyusutan, skenario

##### 2. Tabel Aset Tetap

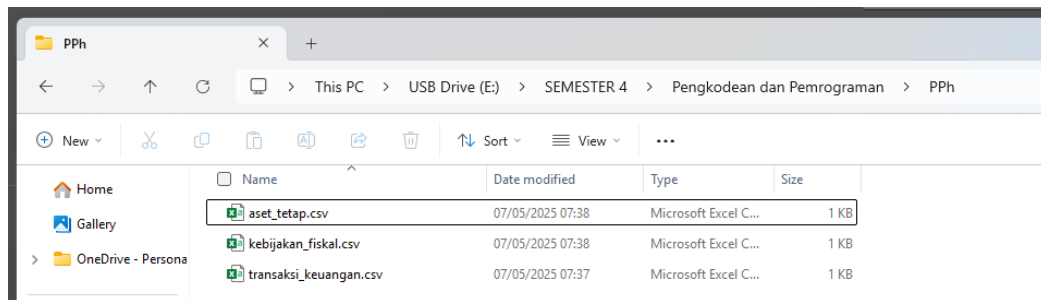
Kolom: aset\_id, kategori, nilai\_perolehan, umur\_ekonomis, metode

##### 3. Tabel Kebijakan Fiskal

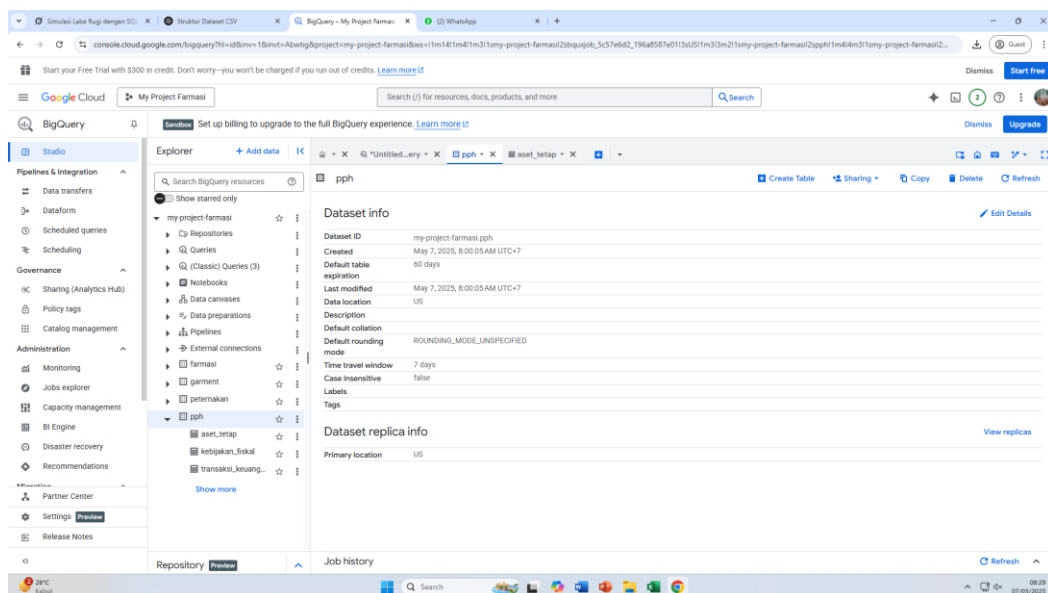
Kolom: tahun, tax\_rate, tax\_holiday\_awal, tax\_holiday\_akhir

#### B. Tahapan Pembuatan Dataset di Google BiQuery

##### 1. Membuat file 3 CSV untuk dengan struktur tabel sesuai struktur dataset di atas.



##### 2. Membuat dataset di Google BigQuery.



### 3. Membuat tabel dengan mengunggah file CSV yang telah dibuat.

#### a. Tabel Transaksi Keuangan

The screenshot shows the Google Cloud BigQuery console interface. The left sidebar displays the project hierarchy under 'My Project Farmasi', with 'transaksi\_keu...' selected. The main panel shows the 'Preview' tab for the 'transaksi\_keu...' table. The table has 3 rows and 7 columns: 'tahun', 'pendapatan', 'beban\_operasi', 'penyusutan', 'skenario', 'tax\_holiday', and 'tax\_holiday'.

Row	tahun	pendapatan	beban_operasi	penyusutan	skenario	tax_holiday
1	2022	100000	60000	10000	normal	
2	2023	120000	65000	12000	tax holiday	
3	2024	90000	50000	9500	tax holiday	

#### b. Tabel Aset Tetap

The screenshot shows the Google Cloud BigQuery console interface. The left sidebar displays the project hierarchy under 'My Project Farmasi', with 'aset\_tetap' selected. The main panel shows the 'Preview' tab for the 'aset\_tetap' table. The table has 3 rows and 7 columns: 'aset\_id', 'kategori', 'nilai\_perolehan', 'umur\_ekonomis', 'metode', 'garis\_lurus', and 'saldo\_menurun'.

Row	aset_id	kategori	nilai_perolehan	umur_ekonomis	metode	garis_lurus
1	A001	Mesin	500000	10	garis_lurus	
2	A003	Bangunan	1000000	20	garis_lurus	
3	A002	Kendaraan	200000	5	saldo_menurun	

#### c. Tabel Kebijakan Fiskal

The screenshot shows the Google Cloud BigQuery console interface. The left sidebar displays the project hierarchy under 'My Project Farmasi', with 'kebijakan\_fiskal' selected. The main panel shows the 'Preview' tab for the 'kebijakan\_fiskal' table. The table has 3 rows and 6 columns: 'tahun', 'tax\_rate', 'tax\_holiday\_awal', 'tax\_holiday\_akhir', 'tax\_holiday\_akhir', and 'tax\_holiday\_akhir'.

Row	tahun	tax_rate	tax_holiday_awal	tax_holiday_akhir	tax_holiday_akhir
1	2022	0.22	2023	2024	
2	2023	0.22	2023	2024	
3	2024	0.22	2023	2024	

## BAB III

### Praktikum Simulasi PPh Badan

#### A. Simulasi Laba/Rugi

##### 1. Membuat Query SQL berbasis Google BigQuery untuk mensimulasikan laba/rugi

The screenshot shows the Google Cloud BigQuery console. A query named 'pph' has been executed successfully. The query selects financial data from a table named 't' and calculates tax-related values. The results are displayed in a table with 10 columns: tahun, skenario, pendapatan, beban\_operasional, penyusutan, laba\_kotor, laba\_operasional, effective\_tax\_rate, pajak, and laba\_bersih. The results show data for three years: 2022 (normal), 2023 (tax holiday), and 2024 (tax holiday).

Row	tahun	skenario	pendapatan	beban_operasional	penyusutan	laba_kotor	laba_operasional	effective_tax_rate	pajak	laba_bersih
1	2022	normal	1000000	600000	100000	400000	300000	0.22	66000.0	234000.0
2	2023	tax holiday	1200000	650000	120000	550000	430000	0.0	0.0	430000.0
3	2024	tax holiday	900000	500000	95000	400000	305000	0.0	0.0	305000.0

#### Query SQL:

WITH LabaRugi AS (

```

SELECT
    t.tahun,
    t.pendapatan,
    t.beban_operasional,
    t.penyusutan,
    t.skenario,
    k.tax_rate,
    k.tax_holiday_awal,
    k.tax_holiday_akhir,
    -- Hitung laba kotor
    (t.pendapatan - t.beban_operasional) AS laba_kotor,
    -- Hitung laba operasional
    (t.pendapatan - t.beban_operasional - t.penyusutan) AS
laba_operasional,
    -- Tentukan apakah tahun ini termasuk dalam tax holiday
    CASE
        WHEN t.tahun BETWEEN k.tax_holiday_awal AND k.tax_holiday_akhir THEN
0
        ELSE k.tax_rate
    END AS effective_tax_rate
FROM
    `my-project-farmasi.pph.transaksi_keuangan` t
LEFT JOIN
    `my-project-farmasi.pph.kebijakan_fiskal` k
ON
    t.tahun = k.tahun
)
SELECT
    tahun,

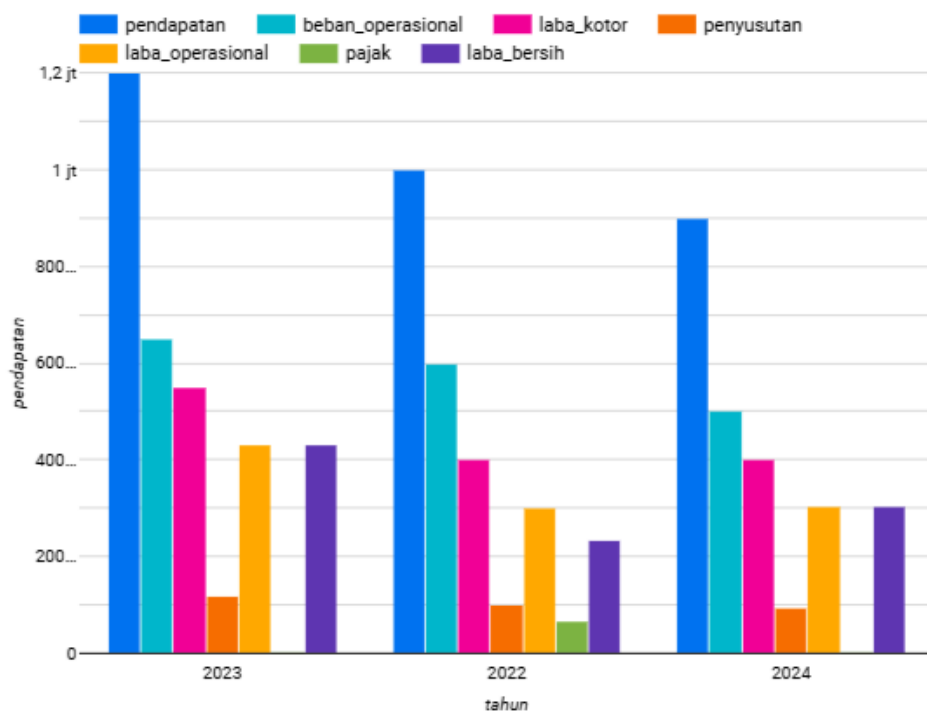
```

```

skenario,
pendapatan,
beban_operasional,
penyusutan,
laba_kotor,
laba_operasional,
effective_tax_rate,
-- Hitung pajak
CASE
  WHEN laba_operasional > 0 THEN laba_operasional * effective_tax_rate
  ELSE 0
END AS pajak,
-- Hitung laba bersih
CASE
  WHEN laba_operasional > 0 THEN laba_operasional * (1 -
effective_tax_rate)
  ELSE laba_operasional
END AS laba_bersih
FROM
  LabaRugi
ORDER BY
  tahun, skenario;

```

## 2. Memvisualisasikan hasil simulasi laba rugi menggunakan Looker studio



	tahun	pendapatan	beban_operasional	laba_kotor	penyusutan	laba_operasional	effective_tax_rate	pajak	laba_bersih
1.	2022	1.000.000	600.000	400.000	100.000	300.000	0,22	66.000	234.000
2.	2023	1.200.000	650.000	550.000	120.000	430.000	0	0	430.000
3.	2024	900.000	500.000	400.000	95.000	305.000	0	0	305.000

Visualisasi berupa diagram batang di atas menunjukkan pendapatan, beban-beban, dan laba dari tahun ke tahun. Secara visual memang sudah menarik, akan tetapi diagram

tersebut kurang informatif. Tidak terdapat angka spesifik dalam satu tingkat/titik data, sehingga masih diperlukan tabel untuk mengetahui tingkat yang pasti dari suatu titik data. Dari tabel tersebut dapat dilihat bahwa pada tahun 2022 tidak diberlakukan skenario *tax holiday* sehingga tarif PPh badan yang dikenakan sebesar 22%. Hal ini sangat berpengaruh terhadap laba bersih tahun 2022 yang terlihat jauh lebih rendah dibanding tahun 2023 dan 2024 yang mana pada kedua tahun tersebut diberlakukan skenario *tax holiday*.

3. Membuat kode python berbasis Google Colab berdasarkan output simulasi laba rugi dari BigQuery untuk membuat visualisasi yang informatif dan menarik

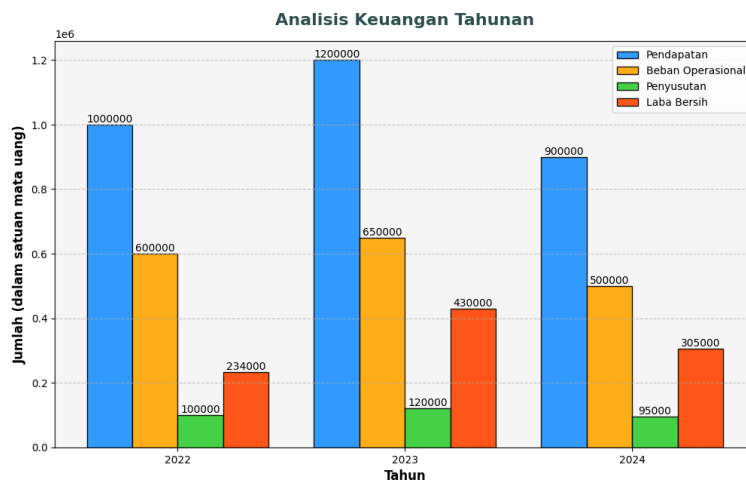


Diagram batang di atas cukup informatif untuk memperbaiki visualisasi sebelumnya. Diagram ini sudah menunjukkan angka spesifik di suatu tingkat/titik data tertentu sehingga lebih mudah dipahami tanpa perlu tabel bantuan.

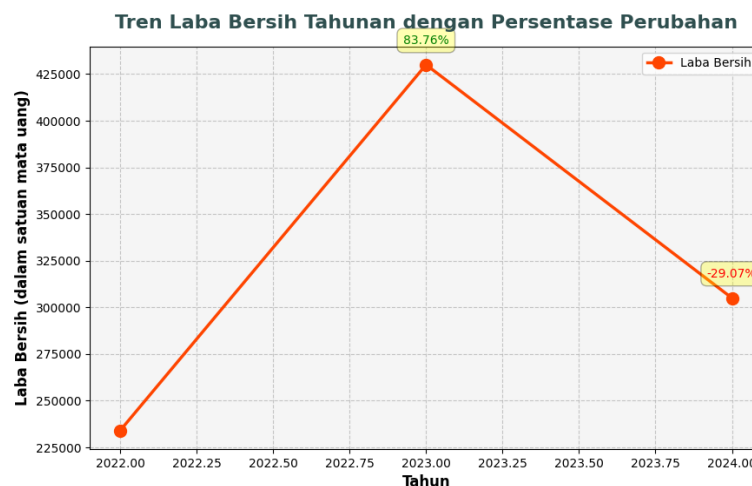


Diagram garis di atas menunjukkan tren perubahan laba bersih dari tahun 2022, 2023, dan 2024. Terlihat bahwa dari tahun 2022 ke tahun 2023 terdapat peningkatan laba bersih yang signifikan sebesar 83,76% sebagai akibat diberlakukannya skenario *tax holiday* yang mana tarif PPh badan sebesar 0%. Sedangkan dari tahun 2023 ke tahun 2024 terdapat penurunan laba bersih yang signifikan sebesar 29,07%, padahal di kedua

tahun ini sama-sama diberlakukan skenario *tax holiday*. Hal ini mengindikasikan bahwa pada tahun 2024 terdapat penurunan kinerja operasional.

### Kode Python:

```
import pandas as pd

import matplotlib.pyplot as plt
import numpy as np

# Data dari tabel Processed Financial Data
data = {
    'tahun': [2022, 2023, 2024],
    'pendapatan': [1000000, 1200000, 900000],
    'beban_operasional': [600000, 650000, 500000],
    'penyusutan': [100000, 120000, 95000],
    'gross_profit': [300000, 430000, 305000],
    'tax': [66000.0, 0.0, 0.0],
    'net_profit': [234000.0, 430000.0, 305000.0]
}
df = pd.DataFrame(data)

# Hitung persentase kenaikan/penurunan laba bersih antar tahun
def calculate_profit_change(df):
    profit_changes = []
    for i in range(1, len(df)):
        prev_profit = df['net_profit'].iloc[i-1]
        curr_profit = df['net_profit'].iloc[i]
        if prev_profit != 0:
            change = ((curr_profit - prev_profit) / prev_profit)
            * 100
        else:
            change = 0
        profit_changes.append(change)
    return profit_changes

# Visualisasi data dengan desain yang lebih menarik
def visualize_data(df, profit_changes):
    years = df['tahun']
    pendapatan = df['pendapatan']
    beban_operasional = df['beban_operasional']
    penyusutan = df['penyusutan']
    net_profit = df['net_profit']

    # Diagram batang dengan desain menarik
    plt.figure(figsize=(12, 7))
    bar_width = 0.2
    index = np.arange(len(years))
```



```

# Warna yang lebih beragam dan efek bayangan
bars1 = plt.bar(index, pendapatan, bar_width,
label='Pendapatan', color='#1E90FF', edgecolor='black',
alpha=0.9)
bars2 = plt.bar(index + bar_width, beban_operasional,
bar_width, label='Beban Operasional', color='#FFA500',
edgecolor='black', alpha=0.9)
bars3 = plt.bar(index + 2 * bar_width, penyusutan, bar_width,
label='Penyusutan', color='#32CD32', edgecolor='black',
alpha=0.9)
bars4 = plt.bar(index + 3 * bar_width, net_profit, bar_width,
label='Laba Bersih', color='#FF4500', edgecolor='black',
alpha=0.9)

# Efek bayangan untuk batang
for bars in [bars1, bars2, bars3, bars4]:
    plt.gca().bar_label(bars, fmt='%.0f', label_type='edge',
color='black')

# Judul dan label dengan gaya modern
plt.xlabel('Tahun', fontsize=12, fontweight='bold')
plt.ylabel('Jumlah (dalam satuan mata uang)', fontsize=12,
fontweight='bold')
plt.title('Analisis Keuangan Tahunan', fontsize=16,
fontweight='bold', pad=15, color='#2F4F4F')
plt.xticks(index + 1.5 * bar_width, years, fontsize=10)
plt.legend(fontsize=10)
plt.grid(True, axis='y', linestyle='--', alpha=0.7)
plt.gca().set_facecolor('#F5F5F5')
plt.show()

# Diagram garis dengan desain menarik
plt.figure(figsize=(10, 6))
plt.plot(years, net_profit, marker='o', color='#FF4500',
linewidth=2.5, markersize=10, label='Laba Bersih')

# Anotasi persentase dengan gaya lebih stylish
for i in range(len(profit_changes)):
    change = profit_changes[i]
    x = years[i+1]
    y = net_profit[i+1]
    color = 'green' if change >= 0 else 'red'
    plt.annotate(f'{change:.2f}%', (x, y), textcoords="offset
points", xytext=(0, 15),
ha='center', va='bottom', fontsize=10,
color=color,
bbox=dict(boxstyle='round,pad=0.5',
fc='yellow', alpha=0.3))

```

```

# Dekorasi tambahan
plt.xlabel('Tahun', fontsize=12, fontweight='bold')
plt.ylabel('Laba Bersih (dalam satuan mata uang)',
fontsize=12, fontweight='bold')
plt.title('Tren Laba Bersih Tahunan dengan Persentase
Perubahan', fontsize=16, fontweight='bold', pad=15,
color='#2F4F4F')
plt.grid(True, linestyle='--', alpha=0.7)
plt.gca().set_facecolor('#F5F5F5')
plt.legend(fontsize=10)
plt.show()

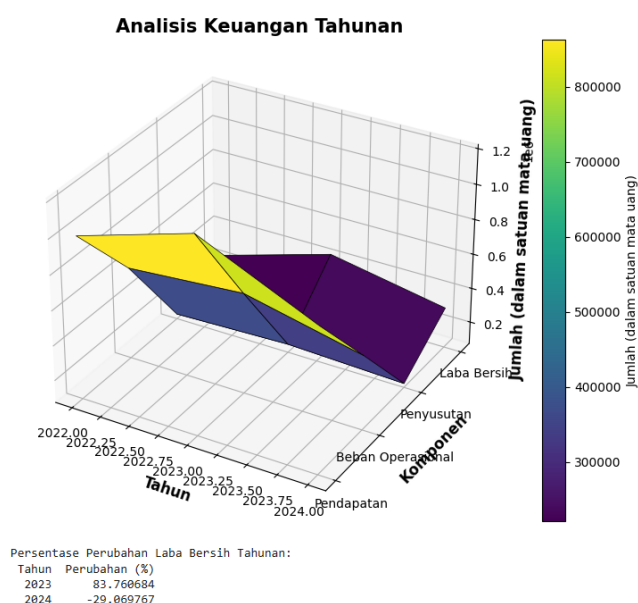
# Main execution
profit_changes = calculate_profit_change(df)

# Tampilkan data dan persentase kenaikan/penurunan
print("Processed Financial Data:")
print(df)
print("\nPersentase Kenaikan/Penurunan Laba Bersih Antar Tahun:")
for i, change in enumerate(profit_changes):
    year_from = df['tahun'].iloc[i]
    year_to = df['tahun'].iloc[i+1]
    print(f"{year_from} ke {year_to}: {change:.2f}%")

# Jalankan visualisasi
visualize_data(df, profit_changes)

```

4. Membuat kode python berbasis Google Colab untuk meningkatkan output menjadi grafik 3 dimensi



Grafik 3 dimensi di atas dapat digunakan untuk menganalisis kondisi keuangan tahunan perusahaan. Namun, hal ini akan menimbulkan kesulitan dalam menentukan tingkat rupiah dalam suatu titik data tertentu karena hanya disajikan dari satu sudut pandang dan dengan skala warna yang sulit dianalisis. Analisis akan mengalami kesulitan untuk mengidentifikasi tingkat pendapatan, beban-beban, dan laba bersih tiap tahun. Sehingga masih diperlukan grafik yang lebih interaktif.

### Kode Python:

```
import pandas as pd

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

# Data keuangan
data = {
    'tahun': [2022, 2023, 2024],
    'pendapatan': [1000000, 1200000, 900000],
    'beban_operasional': [600000, 650000, 500000],
    'penyusutan': [100000, 120000, 95000],
    'gross_profit': [300000, 430000, 305000],
    'tax': [66000.0, 0.0, 0.0],
    'net_profit': [234000.0, 430000.0, 305000.0]
}
df = pd.DataFrame(data)

# Fungsi untuk menghitung persentase perubahan laba bersih
def calculate_profit_change(df):
    profit_changes = []
    for i in range(1, len(df)):
        prev_profit = df['net_profit'].iloc[i-1]
        curr_profit = df['net_profit'].iloc[i]
        change = ((curr_profit - prev_profit) / prev_profit) *
100 if prev_profit != 0 else 0
        profit_changes.append(change)
    return profit_changes

# Fungsi untuk membuat visualisasi 3D
def visualize_data(df, profit_changes):
    years = df['tahun']
    pendapatan = df['pendapatan']
    beban_operasional = df['beban_operasional']
    penyusutan = df['penyusutan']
    net_profit = df['net_profit']

    # Grafik 3D untuk analisis keuangan tahunan
    fig = plt.figure(figsize=(12, 7))
    ax = fig.add_subplot(111, projection='3d')
```

```

# Persiapan data untuk plot surface
x = np.array(years)
y = np.arange(4) # Komponen: 0=Pendapatan, 1=Beban
Operasional, 2=Penyusutan, 3=Laba Bersih
X, Y = np.meshgrid(x, y)
Z = np.array([pendapatan, beban_operasional, penyusutan,
net_profit])

# Plot surface dengan warna gradasi
surf = ax.plot_surface(X, Y, Z, cmap='viridis',
edgecolor='black', linewidth=0.5)
fig.colorbar(surf, label='Jumlah (dalam satuan mata uang)')

# Pengaturan label dan gaya
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

# Data keuangan
data = {
    'tahun': [2022, 2023, 2024],
    'pendapatan': [1000000, 1200000, 900000],
    'beban_operasional': [600000, 650000, 500000],
    'penyusutan': [100000, 120000, 95000],
    'gross_profit': [300000, 430000, 305000],
    'tax': [66000.0, 0.0, 0.0],
    'net_profit': [234000.0, 430000.0, 305000.0]
}
df = pd.DataFrame(data)
# Fungsi untuk menghitung persentase perubahan laba bersih
def calculate_profit_change(df):
    profit_changes = []
    for i in range(1, len(df)):
        prev_profit = df['net_profit'].iloc[i-1]
        curr_profit = df['net_profit'].iloc[i]
        change = ((curr_profit - prev_profit) / prev_profit) *
100 if prev_profit != 0 else 0
        profit_changes.append(change)
    return profit_changes

# Fungsi untuk membuat visualisasi 3D
def visualize_data(df, profit_changes):
    years = df['tahun']
    pendapatan = df['pendapatan']
    beban_operasional = df['beban_operasional']
    penyusutan = df['penyusutan']

```

```

net_profit = df['net_profit']

# Grafik 3D untuk analisis keuangan tahunan
fig = plt.figure(figsize=(12, 7))
ax = fig.add_subplot(111, projection='3d')

# Persiapan data untuk plot surface
x = np.array(years)
y = np.arange(4) # Komponen: 0=Pendapatan, 1=Beban
Operasional, 2=Penyusutan, 3=Laba Bersih
X, Y = np.meshgrid(x, y)
Z = np.array([pendapatan, beban_operasional, penyusutan,
net_profit])

# Plot surface dengan warna gradasi
surf = ax.plot_surface(X, Y, Z, cmap='viridis',
edgecolor='black', linewidth=0.5)
fig.colorbar(surf, label='Jumlah (dalam satuan mata uang)')

# Pengaturan label dan gaya
ax.set_xlabel('Tahun', fontsize=12, fontweight='bold')
ax.set_ylabel('Komponen', fontsize=12, fontweight='bold')
ax.set_zlabel('Jumlah (dalam satuan mata uang)', fontsize=12,
fontweight='bold')

# Pengaturan ticks Y
ax.set_yticks(y)
ax.set_yticklabels(['Pendapatan', 'Beban Operasional',
'Penyusutan', 'Laba Bersih'], fontsize=10)

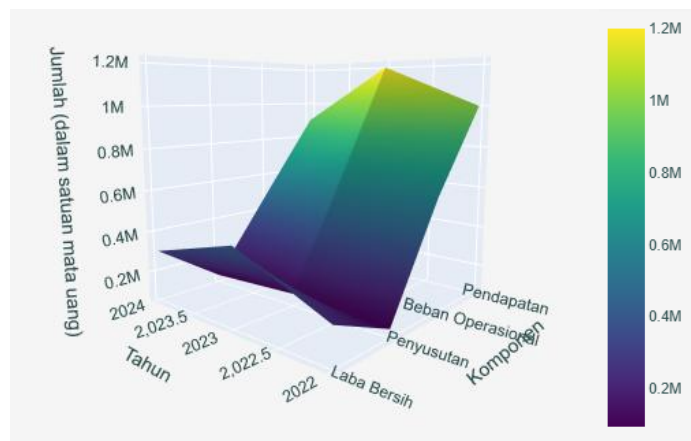
ax.set_title('Analisis Keuangan Tahunan', fontsize=15,
fontweight='bold')
plt.show()

# Tabel persentase perubahan laba bersih
print("\nPersentase Perubahan Laba Bersih Tahunan:")
change_years = df['tahun'].iloc[1:].tolist()
change_data = {'Tahun': change_years, 'Perubahan (%)':
profit_changes}
change_df = pd.DataFrame(change_data)
print(change_df.to_string(index=False))

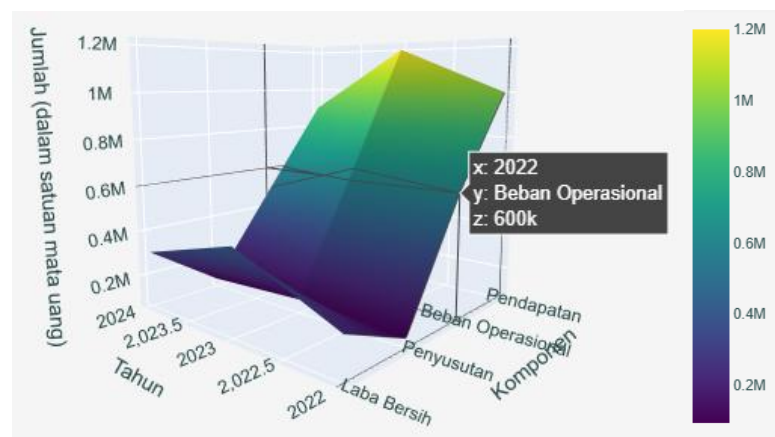
# Eksekusi fungsi
profit_changes = calculate_profit_change(df)
visualize_data(df, profit_changes)

```

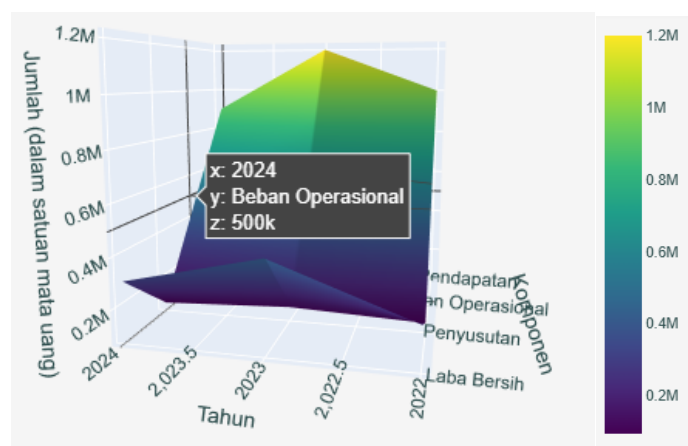
5. Membuat kode python berbasis Google Colab untuk meningkatkan grafik 3 dimensi agar lebih interaktif



Grafik 3 dimensi di atas merupakan grafik yang lebih interaktif (perbaikan dari grafik 3 dimensi sebelumnya). Grafik ini dapat dilihat dari berbagai sisi, dapat diputar, dan menunjukkan tingkat rupiah spesifik dalam tingkat data (titik data) tertentu, sehingga memudahkan analisis untuk menganalisis kondisi keuangan perusahaan. Misalnya:



Dari grafik di atas dapat diketahui bahwa pada tahun 2022, beban operasional perusahaan sebesar Rp 600.000,00.



Dari grafik di atas dapat diketahui bahwa pada tahun 2024, beban operasional perusahaan sebesar Rp 500.000,00.

## Kode Python:

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

# Data keuangan
data = {
    'tahun': [2022, 2023, 2024],
    'pendapatan': [1000000, 1200000, 900000],
    'beban_operasional': [600000, 650000, 500000],
    'penyusutan': [100000, 120000, 95000],
    'gross_profit': [300000, 430000, 305000],
    'tax': [66000.0, 0.0, 0.0],
    'net_profit': [234000.0, 430000.0, 305000.0]
}
df = pd.DataFrame(data)

# Fungsi untuk menghitung persentase perubahan laba bersih
def calculate_profit_change(df):
    profit_changes = []
    for i in range(1, len(df)):
        prev_profit = df['net_profit'].iloc[i-1]
        curr_profit = df['net_profit'].iloc[i]
        change = ((curr_profit - prev_profit) / prev_profit) *
100 if prev_profit != 0 else 0
        profit_changes.append(change)
    return profit_changes

# Fungsi untuk membuat visualisasi 3D interaktif
def visualize_data(df, profit_changes):
    years = df['tahun']
    pendapatan = df['pendapatan']
    beban_operasional = df['beban_operasional']
    penyusutan = df['penyusutan']
    net_profit = df['net_profit']

    # Grafik 3D untuk analisis keuangan tahunan
    fig1 = go.Figure(data=[
        go.Surface(
            z=np.array([pendapatan, beban_operasional,
            penyusutan, net_profit]),
            x=years,
            y=['Pendapatan', 'Beban Operasional', 'Penyusutan',
            'Laba Bersih'],
            colorscale='Viridis',
            showscale=True
        )
    ])
    ])
```

```

fig1.update_layout(
    title='Analisis Keuangan Tahunan 3D',
    scene=dict(
        xaxis_title='Tahun',
        yaxis_title='Komponen',
        zaxis_title='Jumlah (dalam satuan mata uang)'
    ),
    scene_camera_eye=dict(x=1.87, y=0.88, z=0.64),
    plot_bgcolor='#F5F5F5',
    paper_bgcolor='#F5F5F5',
    font=dict(size=12, color='#2F4F4F', family="Arial, bold")
)

fig1.show()

# Grafik 3D untuk tren laba bersih
fig2 = go.Figure(data=[
    go.Scatter3d(
        x=years,
        y=[0] * len(years), # Dummy axis
        z=net_profit,
        mode='lines+markers',
        marker=dict(size=8, color='#FF4500'),
        line=dict(color='#FF4500', width=4)
    )
])

# Tambahkan anotasi persentase untuk 3D
for i in range(len(profit_changes)):
    change = profit_changes[i]
    color = 'green' if change >= 0 else 'red'
    fig2.add_annotation(
        x=years[i+1],
        y=0, # Dummy y-coordinate
        z=net_profit[i+1],
        text=f'{change:.2f}%',
        showarrow=True,
        arrowhead=2,
        ax=0,
        ay=-40,
        font=dict(size=12, color=color),
        bgcolor='yellow',
        opacity=0.7
    )

fig2.update_layout(

```



```

        title='Tren Laba Bersih Tahunan 3D dengan Persentase
Perubahan',
        scene=dict(
            xaxis_title='Tahun',
            yaxis_title='Dummy Axis',
            zaxis_title='Laba Bersih (dalam satuan mata uang)'
        ),
        scene_camera_eye=dict(x=1.87, y=0.88, z=0.64),
        plot_bgcolor='#F5F5F5',
        paper_bgcolor='#F5F5F5',
        font=dict(size=12, color='#2F4F4F', family="Arial,
bold"),
        showlegend=False
    )

    fig2.show()

# Eksekusi utama
profit_changes = calculate_profit_change(df)

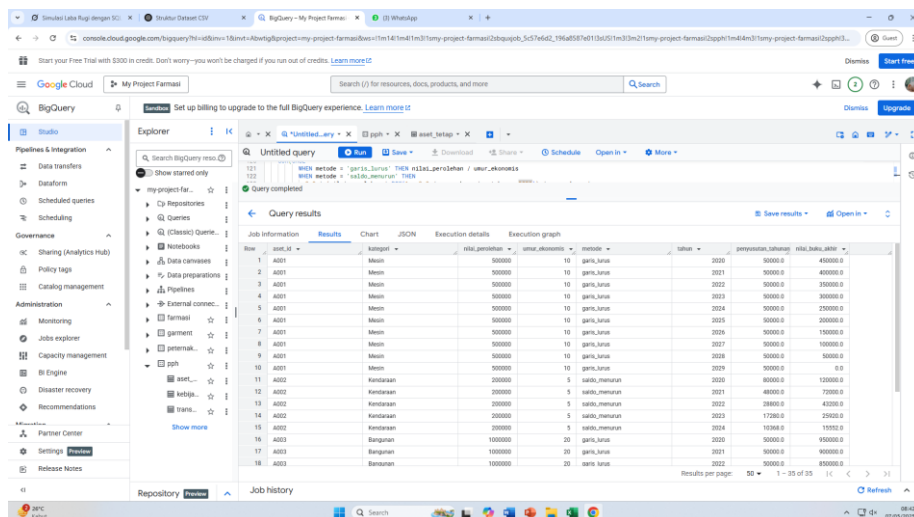
# Tampilkan data dan persentase kenaikan/penurunan
print("Processed Financial Data:")
print(df)
print("\nPersentase Kenaikan/Penurunan Laba Bersih Antar Tahun:")
for i, change in enumerate(profit_changes):
    year_from = df['tahun'].iloc[i]
    year_to = df['tahun'].iloc[i+1]
    print(f"{year_from} ke {year_to}: {change:.2f}%")

# Jalankan visualisasi
visualize_data(df, profit_changes)

```

## B. Simulasi Depresiasi

### 1. Membuat Query SQL berbasis Google BigQuery untuk mensimulasikan depresiasi



The screenshot shows the Google BigQuery Studio interface. The 'Query results' tab is active, displaying a table with 18 rows of data. The table columns are: Row, aset\_id, kategori, nilai\_pemula, persentase\_depresiasi, tahun, and persentase\_sisa. The data shows the depreciation of assets over time, with the remaining value decreasing as the year increases.

Row	aset_id	kategori	nilai_pemula	persentase_depresiasi	tahun	persentase_sisa
1	A001	Mesin	500000	10	2020	450000.0
2	A001	Mesin	500000	10	2021	405000.0
3	A001	Mesin	500000	10	2022	364500.0
4	A001	Mesin	500000	10	2023	328050.0
5	A001	Mesin	500000	10	2024	295245.0
6	A001	Mesin	500000	10	2025	265720.5
7	A001	Mesin	500000	10	2026	239148.5
8	A001	Mesin	500000	10	2027	215233.7
9	A001	Mesin	500000	10	2028	193710.3
10	A001	Mesin	500000	10	2029	174339.3
11	A002	Kendaraan	200000	5	2020	190000.0
12	A002	Kendaraan	200000	5	2021	180500.0
13	A002	Kendaraan	200000	5	2022	171475.0
14	A002	Kendaraan	200000	5	2023	162806.3
15	A002	Kendaraan	200000	5	2024	154466.0
16	A003	Bangunan	1000000	20	2020	800000.0
17	A003	Bangunan	1000000	20	2021	720000.0
18	A003	Bangunan	1000000	20	2022	656000.0

## Query SQL:

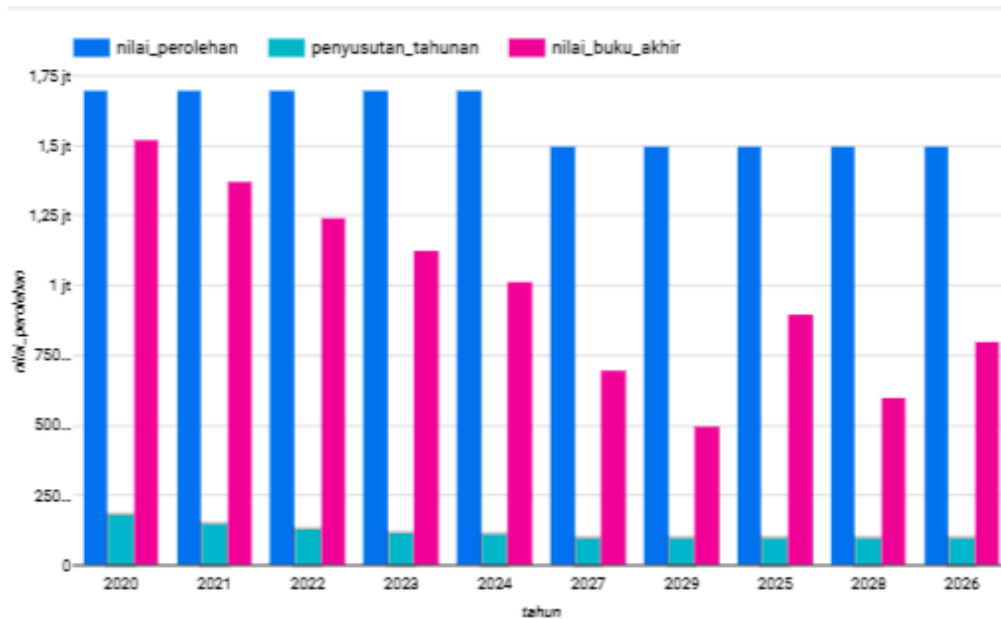
```
WITH AsetDanTahun AS (  
    -- Generate tahun untuk setiap aset berdasarkan umur ekonomis  
    SELECT  
        a.aset_id,  
        a.kategori,  
        a.nilai_perolehan,  
        a.umur_ekonomis,  
        a.metode,  
        tahun  
    FROM  
        `my-project-farmasi.pph.aset_tetap` a  
    CROSS JOIN  
        UNNEST(GENERATE_ARRAY(2020, 2020 + umur_ekonomis - 1)) AS tahun  
),  
PenyusutanTahunan AS (  
    SELECT  
        aset_id,  
        kategori,  
        nilai_perolehan,  
        umur_ekonomis,  
        metode,  
        tahun,  
        CASE  
            WHEN metode = 'garis_lurus' THEN  
                nilai_perolehan / umur_ekonomis  
            WHEN metode = 'saldo_menurun' THEN  
                -- Double declining balance: 2 * (nilai buku awal / umur_ekonomis)  
                2.0 * (  
                    -- Nilai buku awal dihitung berdasarkan akumulasi penyusutan  
                    sebelumnya  
                    nilai_perolehan * POW(1 - 2.0 / umur_ekonomis, tahun - 2020)  
                ) / umur_ekonomis  
            ELSE 0 -- Metode tidak dikenal  
        END AS penyusutan_tahunan,  
        -- Hitung nilai buku akhir  
        CASE  
            WHEN metode = 'garis_lurus' THEN  
                nilai_perolehan - (nilai_perolehan / umur_ekonomis) * (tahun - 2020  
+ 1)  
            WHEN metode = 'saldo_menurun' THEN  
                nilai_perolehan * POW(1 - 2.0 / umur_ekonomis, tahun - 2020 + 1)  
            ELSE nilai_perolehan  
        END AS nilai_buku_akhir  
    FROM  
        AsetDanTahun  
)  
SELECT  
    aset_id,  
    kategori,  
    nilai_perolehan,  
    umur_ekonomis,  
    metode,  
    tahun,  
    ROUND(penyusutan_tahunan, 2) AS penyusutan_tahunan,  
    ROUND(nilai_buku_akhir, 2) AS nilai_buku_akhir
```

```

FROM
    PenyusutanTahunan
WHERE
    -- Pastikan penyusutan tidak negatif dan nilai buku tidak di bawah 0
    penyusutan_tahunan > 0
    AND nilai_buku_akhir >= 0
ORDER BY
    aset_id, tahun;

```

## 2. Memvisualisasikan hasil simulasi depresiasi menggunakan Looker Studio



Grafik di atas menunjukkan nilai perolehan, penyusutan tahunan, dan nilai buku dari aset tetap dalam setiap tahunnya, namun tidak dipisahkan berdasar nama aset tetapnya.

## 3. Membuat kode python berbasis Google Colab berdasarkan file aset tetap CSV untuk membuat simulasi depresiasi untuk tahun 2022, 2023, dan 2024

```

Tabel Depresiasi Aset (2022-2024):
  Aset_ID  Tahun  Kategori  Metode  Depresiasi  Nilai_Buku
0   A001   2022     Mesin   garis_lurus  50000.0   450000.0
1   A002   2022  Kendaraan  saldo_menurun  40000.0   160000.0
2   A003   2022  Bangunan   garis_lurus  50000.0   950000.0
3   A001   2023     Mesin   garis_lurus  50000.0   400000.0
4   A002   2023  Kendaraan  saldo_menurun  32000.0   128000.0
5   A003   2023  Bangunan   garis_lurus  50000.0   900000.0
6   A001   2024     Mesin   garis_lurus  50000.0   350000.0
7   A002   2024  Kendaraan  saldo_menurun  25600.0   102400.0
8   A003   2024  Bangunan   garis_lurus  50000.0   850000.0

Tabel Laba Rugi Setelah Penyesuaian Depresiasi:
  tahun  pendapatan  beban_operasional  penyusutan  gross_profit  tax  \
0   2022   1000000      600000      140000      260000  57200.0
1   2023   1200000      650000      132000      418000    0.0
2   2024    900000      500000      125600      274400    0.0

  net_profit
0   202800.0
1   418000.0
2   274400.0

```

## Kode Python:

```
import pandas as pd

# Data aset tetap (dari aset_tetap.csv)
data_aset = {
    'aset_id': ['A001', 'A002', 'A003'],
    'kategori': ['Mesin', 'Kendaraan', 'Bangunan'],
    'nilai_perolehan': [500000, 200000, 1000000],
    'umur_ekonomis': [10, 5, 20],
    'metode': ['garis_lurus', 'saldo_menurun', 'garis_lurus']
}
df_aset = pd.DataFrame(data_aset)

# Data laba rugi (dari transaksi_keuangan.csv, sebelum
penyesuaian depresiasi)
data_laba_rugi = {
    'tahun': [2022, 2023, 2024],
    'pendapatan': [1000000, 1200000, 900000],
    'beban_operasional': [600000, 650000, 500000],
    'penyusutan': [100000, 120000, 95000], # Akan diganti dengan
perhitungan baru
    'gross_profit': [300000, 430000, 305000],
    'tax': [66000.0, 0.0, 0.0],
    'net_profit': [234000.0, 430000.0, 305000.0]
}
df_laba_rugi = pd.DataFrame(data_laba_rugi)

# Fungsi untuk menghitung depresiasi mulai tahun 2022
def calculate_depreciation(df_aset, start_year=2022,
end_year=2024):
    results = []
    nilai_buku_dict = {row['aset_id']: row['nilai_perolehan'] for
_, row in df_aset.iterrows()}

    for year in range(start_year, end_year + 1):
        year_idx = year - start_year + 1 # Tahun ke-n sejak
mulai
        for idx, row in df_aset.iterrows():
            aset_id = row['aset_id']
            nilai_awal = row['nilai_perolehan']
            umur = row['umur_ekonomis']
            metode = row['metode']
            nilai_buku = nilai_buku_dict[aset_id]

            # Hanya hitung jika masih dalam umur ekonomis
            if year_idx <= umur:
                if metode == 'garis_lurus':
                    depresiasi = nilai_awal / umur
```

```

        else: # saldo_menurun
            depresiasi = nilai_buku * (1 / umur)
            nilai_buku -= depresiasi
            nilai_buku_dict[aset_id] = max(0, nilai_buku) #
Pastikan nilai buku tidak negatif
            results.append([aset_id, year, row['kategori'],
metode, depresiasi, nilai_buku])

    columns = ['Aset_ID', 'Tahun', 'Kategori', 'Metode',
'Depresiasi', 'Nilai_Buku']
    df_results = pd.DataFrame(results, columns=columns)
    return df_results

# Fungsi untuk menyesuaikan laba rugi berdasarkan depresiasi baru
def adjust_profit_loss(df_laba_rugi, df_depresiasi):
    # Hitung total depresiasi per tahun
    total_depresiasi_per_tahun =
df_depresiasi.groupby('Tahun')['Depresiasi'].sum().reset_index()

    # Salin data laba rugi
    df_adjusted = df_laba_rugi.copy()

    # Sesuaikan depresiasi dan hitung ulang laba
    for idx, row in total_depresiasi_per_tahun.iterrows():
        year = row['Tahun']
        new_depresiasi = row['Depresiasi']
        df_adjusted.loc[df_adjusted['tahun'] == year,
'penyusutan'] = new_depresiasi
        df_adjusted.loc[df_adjusted['tahun'] == year,
'gross_profit'] = (
            df_adjusted.loc[df_adjusted['tahun'] == year,
'pendapatan'] -
            df_adjusted.loc[df_adjusted['tahun'] == year,
'beban_operasional'] -
            new_depresiasi
        )
        # Sesuaikan pajak (tax_rate = 0.22 untuk 2022, 0 untuk
2023-2024 karena tax holiday)
        tax_rate = 0.22 if year == 2022 else 0.0
        df_adjusted.loc[df_adjusted['tahun'] == year, 'tax'] = (
            df_adjusted.loc[df_adjusted['tahun'] == year,
'gross_profit'] * tax_rate
        )
        df_adjusted.loc[df_adjusted['tahun'] == year,
'net_profit'] = (
            df_adjusted.loc[df_adjusted['tahun'] == year,
'gross_profit'] -
            df_adjusted.loc[df_adjusted['tahun'] == year, 'tax']

```

```

    )

    return df_adjusted

# Jalankan simulasi depresiasi
depreciation_results = calculate_depreciation(df_aset)

# Sesuaikan laba rugi dengan depresiasi baru
adjusted_laba_rugi = adjust_profit_loss(df_laba_rugi,
depreciation_results)

# Tampilkan hasil
print("Tabel Depresiasi Aset (2022-2024):")
print(depreciation_results)
print("\nTabel Laba Rugi Setelah Penyesuaian Depresiasi:")
print(adjusted_laba_rugi)

```

4. Membuat kode python berbasis Google Colab berdasarkan output simulasi depresiasi untuk membuat visualisasi depresiasi kendaraan

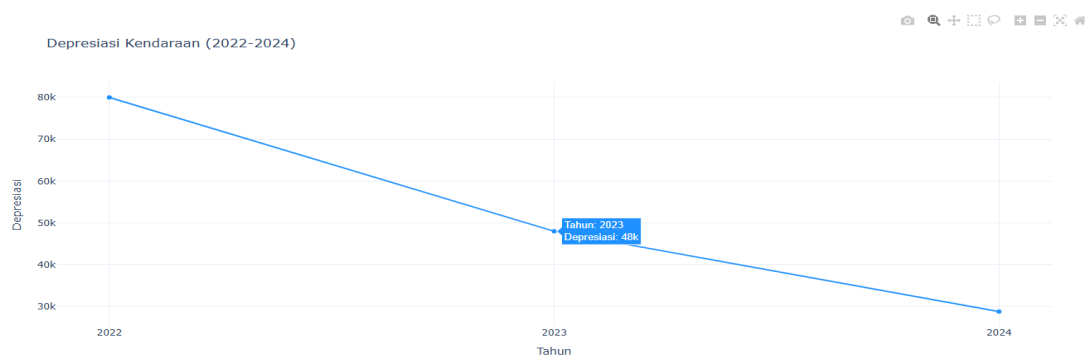


Diagram garis di atas menunjukkan besarnya beban depresiasi aset tetap kendaraan dari tahun 2022, 2023, dan 2024. Kendaraan disusutkan dengan metode menurun berganda (*double declining method*) sehingga terlihat bahwa pada tahun 2022 beban depresiasinya tinggi dan semakin lama semakin rendah.

### Kode Python:

```

import pandas as pd
import plotly.graph_objects as go

# Simulated data loading (Kendaraan only)
data = {
    'aset_id': ['A002'],
    'kategori': ['Kendaraan'],
    'nilai_perolehan': [200000],
    'umur_ekonomis': [5],
    'metode': ['saldo_menurun']
}

```

```

df = pd.DataFrame(data)

# Initialize depreciation data
years = [2022, 2023, 2024]
depreciation_data = []

# Calculate depreciation for Kendaraan (saldo menurun)
for index, row in df.iterrows():
    kategori = row['kategori']
    nilai_awal = row['nilai_perolehan']
    umur = row['umur_ekonomis']
    metode = row['metode']
    nilai_buku = nilai_awal

    for year in years:
        rate = 2 / umur # Double declining balance rate
        depresiasi = nilai_buku * rate
        nilai_buku -= depresiasi # Update nilai_buku for next
year
        depreciation_data.append([year, kategori, depresiasi])

# Create DataFrame
df_depreciation = pd.DataFrame(depreciation_data,
columns=['Tahun', 'Kategori', 'Depresiasi'])

# Create interactive line chart for Kendaraan
fig = go.Figure()
fig.add_trace(go.Scatter(
    x=df_depreciation['Tahun'],
    y=df_depreciation['Depresiasi'],
    mode='lines+markers',
    name='Kendaraan',
    line=dict(color='#1E90FF'),
    hovertemplate='Tahun: %{x}<br>Depresiasi:
%{y}<extra></extra>'
))

# Update layout
fig.update_layout(
    title='Depresiasi Kendaraan (2022-2024)',
    xaxis_title='Tahun',
    yaxis_title='Depresiasi',
    xaxis=dict(tickmode='array', tickvals=[2022, 2023, 2024]),
    hovermode='closest',
    template='plotly_white'
)

# Show the plot

```

```
fig.show()
```

5. Membuat kode python berbasis Google Colab berdasarkan output simulasi depresiasi untuk membuat visualisasi depresiasi mesin

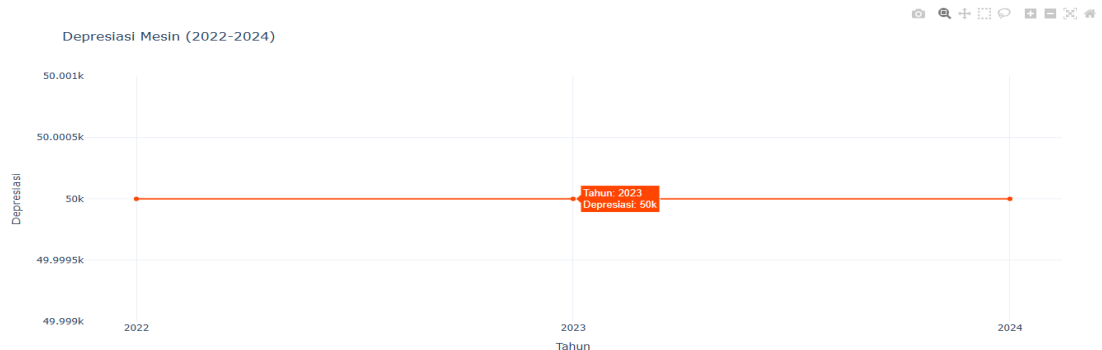


Diagram garis di atas menunjukkan besarnya beban depresiasi aset tetap mesin dari tahun 2022, 2023, dan 2024. Mesin disusutkan dengan metode garis lurus (*straight line method*) sehingga terlihat bahwa besarnya beban depresiasi pada setiap tahunnya tetap sama (konstan).

### Kode Python:

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

# Simulated data loading (Mesin only)
data = {
    'aset_id': ['A001'],
    'kategori': ['Mesin'],
    'nilai_perolehan': [500000],
    'umur_ekonomis': [10],
    'metode': ['garis_lurus']
}
df = pd.DataFrame(data)

# Initialize depreciation data
years = [2022, 2023, 2024]
depreciation_data = []

# Calculate depreciation for Mesin (garis lurus)
for index, row in df.iterrows():
    kategori = row['kategori']
    nilai_awal = row['nilai_perolehan']
    umur = row['umur_ekonomis']
    metode = row['metode']
    nilai_buku = nilai_awal

    for year in years:
```



```

        depresiasi = nilai_awal / umur
        nilai_buku -= depresiasi # Update nilai_buku for next
year
        depreciation_data.append([year, kategori, depresiasi])

# Create DataFrame
df_depreciation = pd.DataFrame(depreciation_data,
columns=['Tahun', 'Kategori', 'Depresiasi'])

# Create interactive line chart for Mesin
fig = go.Figure()
fig.add_trace(go.Scatter(
    x=df_depreciation['Tahun'],
    y=df_depreciation['Depresiasi'],
    mode='lines+markers',
    name='Mesin',
    line=dict(color='#FF4500'),
    hovertemplate='Tahun: %{x}<br>Depresiasi:
%{y}<extra></extra>'
))

# Update layout
fig.update_layout(
    title='Depresiasi Mesin (2022-2024)',
    xaxis_title='Tahun',
    yaxis_title='Depresiasi',
    xaxis=dict(tickmode='array', tickvals=[2022, 2023, 2024]),
    hovermode='closest',
    template='plotly_white'
)

# Show the plot
fig.show()

```

6. Membuat kode python berbasis Google Colab berdasarkan output simulasi depresiasi untuk membuat visualisasi depresiasi bangunan

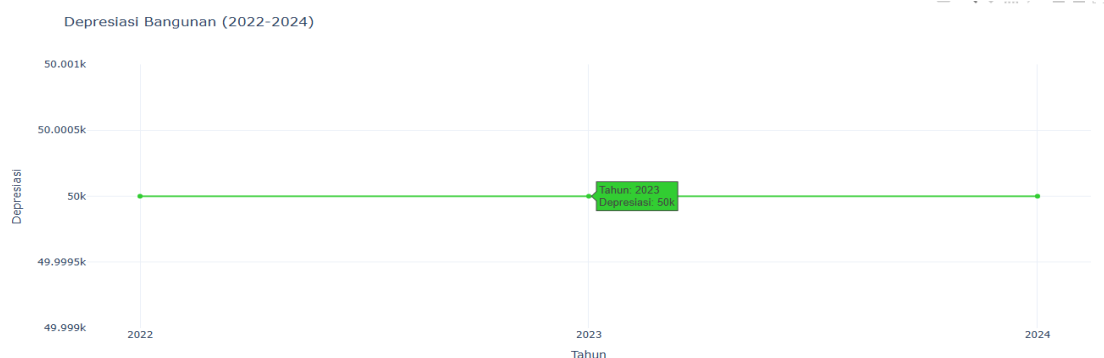


Diagram garis di atas menunjukkan besarnya beban depresiasi aset tetap mesin dari tahun 2022, 2023, dan 2024. Mesin disusutkan dengan metode garis lurus (*straight line method*) sehingga terlihat bahwa besarnya beban depresiasi pada setiap tahunnya tetap sama (konstan).

### Kode Python:

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

# Simulated data loading (Bangunan only)
data = {
    'aset_id': ['A003'],
    'kategori': ['Bangunan'],
    'nilai_perolehan': [1000000],
    'umur_ekonomis': [20],
    'metode': ['garis_lurus']
}
df = pd.DataFrame(data)

# Initialize depreciation data
years = [2022, 2023, 2024]
depreciation_data = []

# Calculate depreciation for Bangunan (garis lurus)
for index, row in df.iterrows():
    kategori = row['kategori']
    nilai_awal = row['nilai_perolehan']
    umur = row['umur_ekonomis']
    metode = row['metode']
    nilai_buku = nilai_awal

    for year in years:
        depresiasi = nilai_awal / umur
        nilai_buku -= depresiasi # Update nilai_buku for next year
        depreciation_data.append([year, kategori, depresiasi])

# Create DataFrame
df_depreciation = pd.DataFrame(depreciation_data,
                                columns=['Tahun', 'Kategori', 'Depresiasi'])

# Create interactive line chart for Bangunan
fig = go.Figure()
fig.add_trace(go.Scatter(
    x=df_depreciation['Tahun'],
    y=df_depreciation['Depresiasi'],
    mode='lines+markers',
```

```

        name='Bangunan',
        line=dict(color='#32CD32'),
        hovertemplate='Tahun: %{x}<br>Depresiasi:
        %{y}<extra></extra>'
    ))

# Update layout
fig.update_layout(
    title='Depresiasi Bangunan (2022-2024)',
    xaxis_title='Tahun',
    yaxis_title='Depresiasi',
    xaxis=dict(tickmode='array', tickvals=[2022, 2023, 2024]),
    hovermode='closest',
    template='plotly_white'
)

# Show the plot
fig.show()

```

## C. Hasil Analisis

### 1. Skenario *Tax Holiday*

Tax holiday merupakan insentif pajak yang diberikan oleh pemerintah kepada perusahaan, biasanya dalam bentuk pembebasan atau pengurangan pajak penghasilan (PPh) badan untuk jangka waktu tertentu. Insentif ini umumnya ditujukan untuk menarik investasi, terutama di sektor-sektor prioritas atau wilayah tertentu. Pemberian tax holiday dapat berdampak signifikan terhadap laba bersih perusahaan, karena langsung memengaruhi jumlah pajak yang harus dibayarkan, sehingga meningkatkan keuntungan setelah pajak.

Dengan diberikannya tax holiday, perusahaan tidak perlu membayar atau hanya membayar sebagian kecil dari kewajiban PPh Badan selama periode tertentu. Hal ini menyebabkan pengeluaran pajak berkurang drastis, sehingga komponen beban pajak dalam laporan laba rugi menjadi lebih kecil. Dengan berkurangnya beban pajak, maka laba sebelum pajak dan laba bersih perusahaan akan semakin mendekati nilai yang sama. Kenaikan laba bersih ini bisa digunakan untuk memperkuat struktur modal, memperluas usaha, atau meningkatkan dividen bagi pemegang saham, tergantung pada kebijakan perusahaan.

Dalam simulasi ini, diberlakukan tax holiday pada tahun 2023 dan 2024, sehingga laba bersih tahun 2022 jauh lebih rendah dibandingkan tahun 2023 dan 2024 karena perusahaan harus membayar PPh badan sebesar 22% . Pada tahun 2023 terdapat

kenaikan laba bersih secara signifikan karena diberlakukannya tax holiday yang mana tarif PPh sebesar 0%. Sementara itu, pada tahun 2024 yang sama-sama diberlakukan tax holiday, terdapat penurunan signifikan pada laba bersih yang mengindikasikan adanya penurunan kinerja perusahaan.

## **2. Metode Penyusutan**

Metode penyusutan aktiva tetap, seperti garis lurus (*straight-line*) dan menurun berganda (*double declining balance*), memiliki pengaruh langsung terhadap laba bersih perusahaan, terutama karena perbedaan besarnya beban penyusutan yang dicatat tiap periode. Beban penyusutan merupakan komponen biaya yang mengurangi laba, sehingga metode yang digunakan dalam perhitungannya akan memengaruhi besar kecilnya laba sebelum pajak dan laba bersih.

Pada metode garis lurus, beban penyusutan dibagi secara merata sepanjang umur manfaat aset. Artinya, perusahaan akan mencatat beban penyusutan yang sama setiap tahunnya. Hal ini menyebabkan laba bersih cenderung stabil selama periode penggunaan aset, karena tidak ada fluktuasi besar dari sisi biaya penyusutan. Metode ini cocok untuk aset yang nilai manfaat ekonominya dianggap konstan dari tahun ke tahun.

Sebaliknya, metode menurun berganda menghasilkan beban penyusutan yang lebih besar pada tahun-tahun awal penggunaan aset dan lebih kecil pada tahun-tahun berikutnya. Hal ini menyebabkan laba bersih pada awal masa penggunaan aset akan lebih rendah, karena beban penyusutan yang tinggi, sedangkan pada tahun-tahun berikutnya laba akan meningkat seiring berkurangnya beban penyusutan. Metode ini dapat bermanfaat dari sisi perpajakan karena menunda pengakuan laba kena pajak, tetapi juga membuat laporan keuangan tampak kurang menguntungkan pada awalnya. Oleh karena itu, pemilihan metode penyusutan bukan hanya berdampak pada laporan keuangan, tetapi juga dapat memengaruhi strategi perpajakan dan persepsi investor terhadap profitabilitas perusahaan.

## **BAB IV**

### **Penutup**

#### **A. Kesimpulan Penggunaan Platform**

Waktu yang tepat untuk menggunakan Google BigQuery dan Google Colab dalam analisis data tergantung pada jenis data yang dimiliki, kompleksitas analisis, serta kebutuhan kolaborasi dan efisiensi.

##### **1. Google BigQuery**

Google BigQuery adalah layanan data warehouse berbasis cloud yang disediakan oleh Google Cloud Platform (GCP). BigQuery dirancang untuk menyimpan, mengelola, dan menganalisis data dalam jumlah besar (big data) secara cepat dan efisien dengan menggunakan kemampuan komputasi terdistribusi. Layanan ini sangat cocok untuk analisis data skala besar karena dapat menangani miliaran baris data dan menjalankan kueri kompleks hanya dalam hitungan detik atau menit.

BigQuery menggunakan bahasa SQL (Structured Query Language) untuk memudahkan pengguna dalam menulis kueri dan menganalisis data. Pengguna tidak perlu mengatur infrastruktur server atau database secara manual, karena BigQuery bersifat serverless, yang berarti Google secara otomatis mengelola skalabilitas, performa, dan pemeliharaan sistem di belakang layar. Ini sangat memudahkan para analis, peneliti, atau mahasiswa yang ingin fokus pada analisis tanpa harus repot dengan aspek teknis penyimpanan data. BigQuery cocok digunakan untuk:

- Analisis data skala besar seperti log transaksi, data keuangan historis, atau data perpajakan dari banyak entitas.
- Proyek yang membutuhkan kecepatan query tinggi dan efisiensi biaya (karena hanya membayar data yang diproses).
- Integrasi dengan berbagai sumber data melalui Google Cloud Platform.
- Situasi di mana data disimpan dalam cloud dan butuh keamanan serta skalabilitas.

##### **2. Google Colab**

Google Colab (kepanjangan dari Google Colaboratory) adalah platform berbasis cloud yang disediakan oleh Google untuk menulis dan menjalankan kode Python langsung dari browser, tanpa perlu instalasi atau konfigurasi perangkat lunak tambahan di komputer lokal. Google Colab sangat populer di kalangan pelajar, peneliti,

dan praktisi data science karena menyediakan lingkungan pemrograman yang gratis, interaktif, dan mudah digunakan.

Salah satu kekuatan utama Google Colab adalah kemampuannya untuk menjalankan kode Python dengan dukungan GPU dan TPU secara gratis, yang sangat berguna untuk analisis data skala besar, machine learning, dan deep learning. Colab juga mendukung berbagai pustaka Python populer seperti Pandas, NumPy, Matplotlib, Scikit-learn, TensorFlow, dan banyak lagi, sehingga sangat cocok untuk eksplorasi data, visualisasi, hingga pembuatan model prediktif. Google Colab cocok digunakan untuk:

- Ingin menulis dan menjalankan kode Python untuk analisis data, visualisasi, machine learning, atau simulasi.
- Butuh lingkungan yang gratis, mudah diakses via browser, dan mendukung pustaka Python seperti Pandas, Matplotlib, Scikit-learn, dan TensorFlow.
- Melakukan eksperimen data dan skrip yang bersifat interaktif.
- Membutuhkan kolaborasi dengan tim atau dosen, karena notebook dapat dibagikan dan disunting bersama.

### **3. Kombinasi Google BigQuery dan Google Colab**

Kombinasi ini cocok digunakan untuk:

- Mengambil data besar dari BigQuery, lalu memproses atau memvisualisasikannya di Google Colab.
- Membuat simulasi perpajakan, prediksi tren keuangan, atau analisis historis dengan data yang disimpan di cloud.
- Mengotomasi pipeline analisis data dari pengambilan, pengolahan, sampai pelaporan dalam satu workflow