

TUGAS AKHIR

MATA KULIAH PENGKODEAN DAN PEMROGRAMAN

KELAS E

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



Disusun oleh:

Vinny Anindhia Khansa
12030123140243

PROGRAM STUDI AKUNTANSI

FAKULTAS EKONOMIKA DAN BISNIS

UNIVERSITAS DIPONEGORO

2025

BAB I. BIG QUERY

1.1. Visualisasi Data dan Grafik

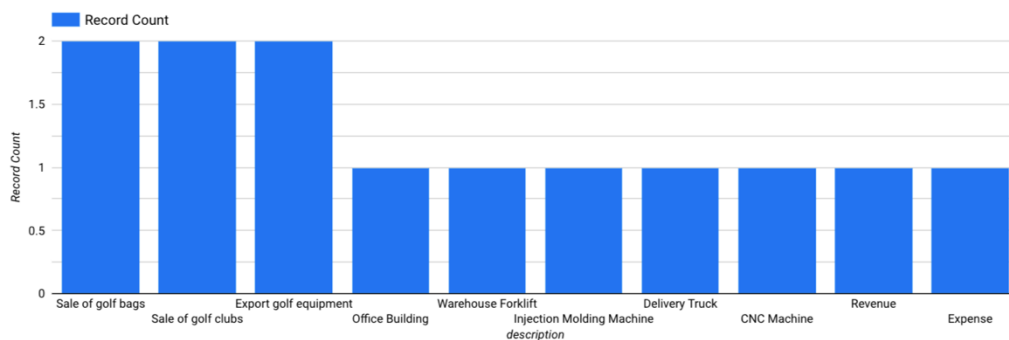
Hasil Query

Query results								
<div>Save results Open in</div>								
Job information Results Chart JSON Execution details Execution graph								
Row	simulation_type	id	description	metric_1	metric_2	metric_3	tax_amount	
1	Depreciation Simulation	AST001	Injection Molding Machine	50000000.0	100000000.0	400000000.0	null	
2	Depreciation Simulation	AST002	Delivery Truck	40000000.0	80000000.0	160000000.0	null	
3	Depreciation Simulation	AST003	Office Building	50000000.0	100000000.0	900000000.0	null	
4	Depreciation Simulation	AST004	CNC Machine	37500000.0	75000000.0	262500000.0	null	
5	Depreciation Simulation	AST005	Warehouse Forklift	16000000.0	32000000.0	68000000.0	null	
6	Normal Simulation	null	Revenue	3.0	48000000.0	16000000.0	null	
7	Normal Simulation	null	Expense	2.0	-7000000.0	-3500000.0	null	
8	Tax Holiday Simulation	TRX001	Sale of golf clubs	15000000.0	25.0	null	3750000.0	
9	Tax Holiday Simulation	TRX001	Sale of golf clubs	15000000.0	0.0	null	0.0	
10	Tax Holiday Simulation	TRX003	Sale of golf bags	8000000.0	25.0	null	2000000.0	
11	Tax Holiday Simulation	TRX003	Sale of golf bags	8000000.0	0.0	null	0.0	
12	Tax Holiday Simulation	TRX005	Export golf equipment	25000000.0	25.0	null	6250000.0	
13	Tax Holiday Simulation	TRX005	Export golf equipment	25000000.0	0.0	null	0.0	

BigQuery Custom SQL

simulation_type		Record Count
1.	Tax Holiday Simulation	6
2.	Depreciation Simulation	5
3.	Normal Simulation	2

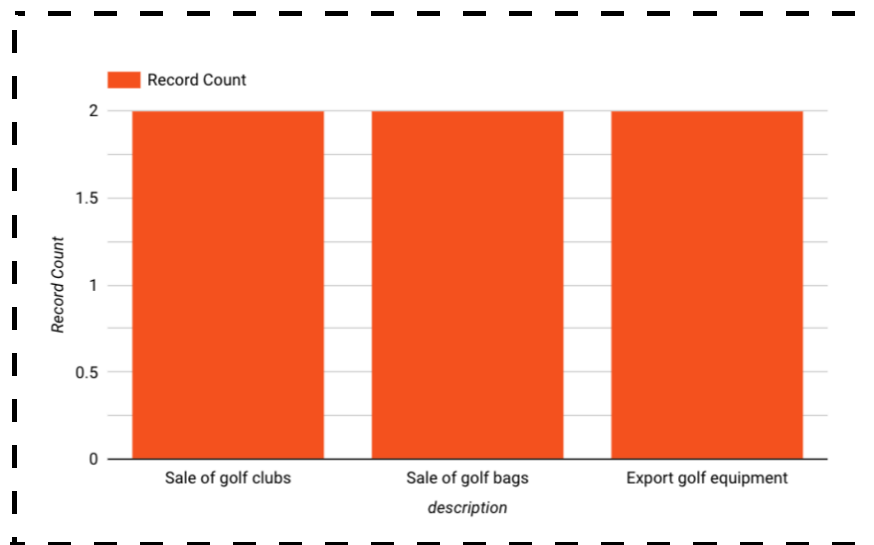
1 - 3 / 3 < >



1.2. Penjelasan

Terdapat tiga hasil yang bisa dijabarkan yaitu:

a. Tax Holiday Simulation



Histogram menunjukkan tiga transaksi pendapatan: "Sale of golf clubs", "Sale of golf bags", dan "Export golf equipment", masing-masing dengan 2 rekord. Ini karena setiap transaksi pendapatan dievaluasi terhadap dua kebijakan pajak (Corporate Income Tax 25% dan Export Tax Incentive 0%).

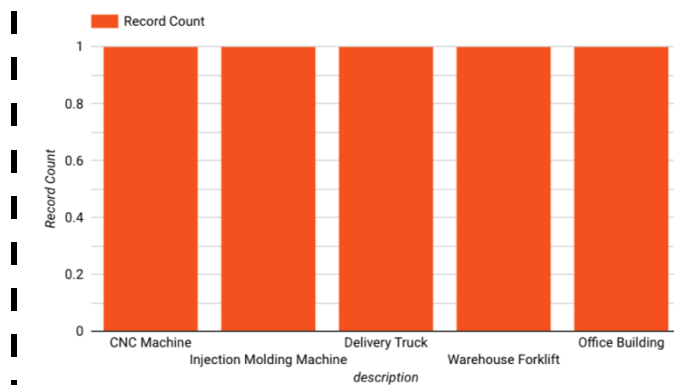
Dampak pada Laba Rugi:

- Transaksi ekspor ("Export golf equipment", 25 juta) mendapat insentif pajak 0%, sehingga meningkatkan laba bersih sebesar 25 juta dibandingkan jika dikenakan pajak 25% (6,25 juta).
- Transaksi domestik ("Sale of golf clubs" 15 juta, "Sale of golf bags" 8 juta) dikenakan pajak 25%, mengurangi laba bersih masing-masing sebesar 3,75 juta dan 2 juta.
- Total penghematan pajak dari insentif ekspor: 6,25 juta, meningkatkan laba bersih.

Dampak pada Modal Kerja:

- Penghematan pajak meningkatkan arus kas, sehingga modal kerja bertambah sebesar 6,25 juta, yang dapat digunakan untuk operasional atau investasi jangka pendek.

a. Depreciation Simulation



Histogram menunjukkan lima aset tetap: "CNC Machine", "Delivery Truck", "Injection Molding Machine", "Warehouse Forklift", dan "Office Building", masing-masing dengan 1 rekord, mencerminkan perhitungan depresiasi untuk setiap aset.

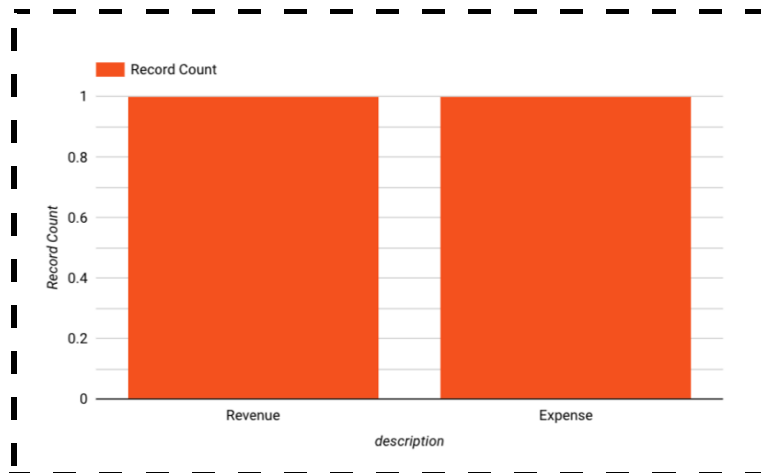
Dampak pada Laba Rugi:

- Depresiasi mengurangi laba kena pajak. Dengan metode garis lurus, total depresiasi tahunan: 50 juta (Injection Molding Machine) + 40 juta (Delivery Truck) + 50 juta (Office Building) + 37,5 juta (CNC Machine) + 16 juta (Warehouse Forklift) = 193,5 juta/tahun.
- Jika metode saldo menurun digunakan, depresiasi lebih tinggi di tahun awal (misalnya, 100 juta untuk Injection Molding Machine), mengurangi laba kena pajak lebih besar di awal, sehingga menghemat pajak lebih banyak di periode awal.

Dampak pada Modal Kerja:

- Depresiasi adalah beban non-kas, jadi tidak langsung memengaruhi arus kas atau modal kerja. Namun, penghematan pajak akibat depresiasi (25% dari 193,5 juta = 48,375 juta) meningkatkan arus kas, yang dapat memperkuat modal kerja.

b. Normal Simulation



Histogram menunjukkan dua kategori transaksi: "Revenue" dan "Expense", masing-masing dengan 1 rekord. Revenue mencakup 3 transaksi (48 juta), dan Expense mencakup 2 transaksi (-7 juta).

Dampak pada Laba Rugi:

- Total pendapatan bersih sebelum pajak: 48 juta - 7 juta = 41 juta.
- Setelah pajak 25% (tanpa insentif ekspor), pajak atas pendapatan adalah 12 juta, sehingga laba bersih menjadi 29 juta.

Dampak pada Modal Kerja:

- Arus kas bersih dari operasional (41 juta sebelum pajak) meningkatkan modal kerja, tetapi pembayaran pajak 12 juta mengurangnya. Modal kerja bersih bertambah sebesar 29 juta, yang dapat digunakan untuk kebutuhan operasional atau investasi.

1.3. Kode

-- Assume tables are created in BigQuery based on the provided CSV structures
 -- Creating temporary tables for demonstration

WITH financial_transactions AS (

```
SELECT
  transaction_id,
  transaction_date,
  category,
  description,
  amount,
  currency,
  account_number,
  payment_method
FROM UNNEST([
```

```

STRUCT('TRX001' AS transaction_id, DATE('2025-01-15') AS
transaction_date, 'Revenue' AS category, 'Sale of golf clubs' AS description,
15000000 AS amount, 'IDR' AS currency, 'ACC123456' AS account_number,
'Credit' AS payment_method),
STRUCT('TRX002', DATE('2025-01-16'), 'Expense', 'Raw material purchase', -
5000000, 'IDR', 'ACC123456', 'Bank Transfer'),
STRUCT('TRX003', DATE('2025-02-01'), 'Revenue', 'Sale of golf bags',
8000000, 'IDR', 'ACC123456', 'Cash'),
STRUCT('TRX004', DATE('2025-02-10'), 'Expense', 'Factory maintenance', -
2000000, 'IDR', 'ACC123456', 'Bank Transfer'),
STRUCT('TRX005', DATE('2025-03-05'), 'Revenue', 'Export golf equipment',
25000000, 'IDR', 'ACC789012', 'Credit')
])
),
fixed_assets AS (
SELECT
asset_id,
asset_name,
category,
purchase_date,
purchase_value,
useful_life_years,
depreciation_method,
current_value
FROM UNNEST([
STRUCT('AST001' AS asset_id, 'Injection Molding Machine' AS asset_name,
'Machinery' AS category, DATE('2023-06-10') AS purchase_date, 500000000 AS
purchase_value, 10 AS useful_life_years, 'Straight-Line' AS
depreciation_method, 400000000 AS current_value),
STRUCT('AST002', 'Delivery Truck', 'Vehicle', DATE('2024-01-15'),
200000000, 5, 'Straight-Line', 160000000),
STRUCT('AST003', 'Office Building', 'Building', DATE('2022-03-01'),
1000000000, 20, 'Straight-Line', 900000000),
STRUCT('AST004', 'CNC Machine', 'Machinery', DATE('2023-09-20'),
300000000, 8, 'Straight-Line', 262500000),
STRUCT('AST005', 'Warehouse Forklift', 'Equipment', DATE('2024-05-05'),
80000000, 5, 'Straight-Line', 68000000)
])
),
fiscal_policies AS (
SELECT
policy_id,
policy_name,
effective_date,
description,
tax_rate,
compliance_status
FROM UNNEST([

```

```

STRUCT('POL001' AS policy_id, 'Corporate Income Tax' AS policy_name,
DATE('2025-01-01') AS effective_date, 'Annual corporate tax obligation' AS
description, 25.0 AS tax_rate, 'Compliant' AS compliance_status),
STRUCT('POL002', 'VAT on Sales', DATE('2025-01-01'), 'Value-added tax on
product sales', 11.0, 'Compliant'),
STRUCT('POL003', 'Import Duty', DATE('2025-01-01'), 'Duty on imported raw
materials', 5.0, 'Compliant'),
STRUCT('POL004', 'Depreciation Policy', DATE('2025-01-01'), 'Asset
depreciation for tax purposes', NULL, 'Compliant'),
STRUCT('POL005', 'Export Tax Incentive', DATE('2025-01-01'), 'Tax relief for
export activities', 0.0, 'Compliant')
])
),

```

-- Normal Simulation: Summarize financial transactions

```

normal_simulation AS (
SELECT
    category,
    COUNT(*) AS transaction_count,
    SUM(amount) AS total_amount,
    AVG(amount) AS avg_amount
FROM financial_transactions
GROUP BY category
),

```

-- Depreciation Simulation: Straight-Line vs Double-Declining Balance

```

depreciation_simulation AS (
SELECT
    asset_id,
    asset_name,
    purchase_value,
    useful_life_years,
    -- Straight-Line Depreciation
    purchase_value / useful_life_years AS straight_line_annual_depreciation,
    -- Double-Declining Balance Depreciation
    (2.0 / useful_life_years) * purchase_value AS
double_declining_annual_depreciation,
    -- Current book value for reference
    current_value
FROM fixed_assets
),

```

-- Tax Holiday Simulation: Apply corporate tax and export tax incentive

```

tax_holiday_simulation AS (
SELECT
    ft.transaction_id,
    ft.description,
    ft.amount,
    fp.policy_name,

```



```

    fp.tax_rate,
    CASE
        WHEN fp.policy_name = 'Export Tax Incentive' AND ft.description LIKE
'%Export%'
        THEN 0.0
        WHEN fp.policy_name = 'Corporate Income Tax' AND ft.category =
'Revenue'
        THEN ft.amount * (fp.tax_rate / 100)
        ELSE 0.0
    END AS tax_amount
FROM financial_transactions ft
CROSS JOIN fiscal_policies fp
WHERE fp.policy_name IN ('Corporate Income Tax', 'Export Tax Incentive')
    AND ft.category = 'Revenue'
)

```

-- Combine results for final output

```

SELECT
    'Normal Simulation' AS simulation_type,
    CAST(NULL AS STRING) AS id,
    category AS description,
    transaction_count AS metric_1,
    total_amount AS metric_2,
    avg_amount AS metric_3,
    CAST(NULL AS FLOAT64) AS tax_amount
FROM normal_simulation
UNION ALL
SELECT
    'Depreciation Simulation' AS simulation_type,
    asset_id AS id,
    asset_name AS description,
    straight_line_annual_depreciation AS metric_1,
    double_declining_annual_depreciation AS metric_2,
    current_value AS metric_3,
    CAST(NULL AS FLOAT64) AS tax_amount
FROM depreciation_simulation
UNION ALL
SELECT
    'Tax Holiday Simulation' AS simulation_type,
    transaction_id AS id,
    description,
    amount AS metric_1,
    tax_rate AS metric_2,
    CAST(NULL AS FLOAT64) AS metric_3,
    tax_amount
FROM tax_holiday_simulation
ORDER BY simulation_type, id;

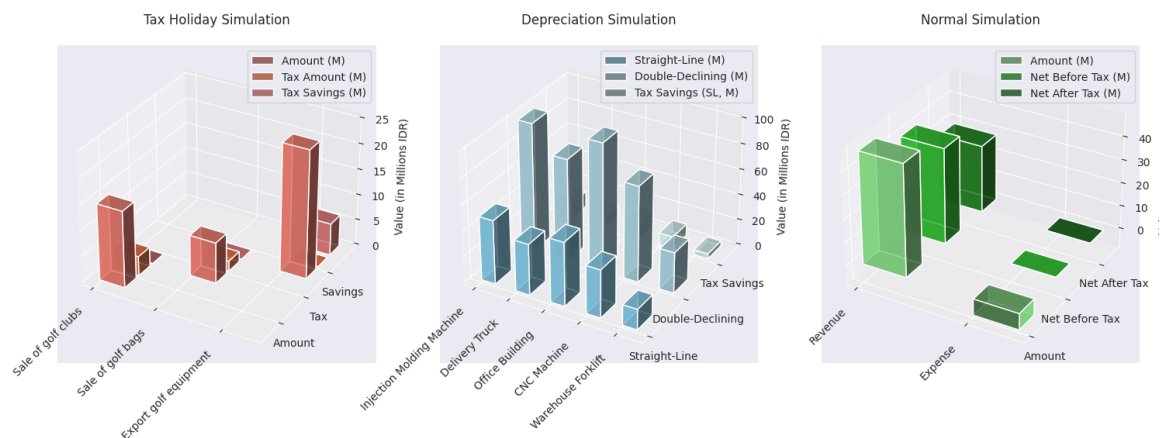
```

BAB II. PHYTON

2.1. Visualisasi Data dan Grafik

Hasil Grafik

Impact of Simulations on GolfHouse Financials



2.2. Penjelasan

Terdapat tiga hasil yang bisa dijabarkan yaitu:

- Tax Holiday Simulation Subplot ini menunjukkan tiga transaksi pendapatan: "Sale of golf clubs" (15 juta), "Sale of golf bags" (8 juta), dan "Export golf equipment" (25 juta).

Metrik:

- Amount (M) (salmon): Nilai transaksi dalam jutaan IDR.
- Tax Amount (M) (coral): Pajak yang dikenakan (25% untuk domestik, 0% untuk ekspor), yaitu 3,75 juta, 2 juta, dan 0 juta.
- Tax Savings (M) (lightcoral): Penghematan pajak dari insentif ekspor, yaitu 6,25 juta untuk "Export golf equipment".

Simulasi ini menyoroti manfaat insentif pajak ekspor, yang meningkatkan laba bersih dan arus kas dengan mengurangi beban pajak pada transaksi ekspor.

- Depreciation Simulation

Subplot ini menampilkan depresiasi untuk lima aset tetap: "Injection Molding Machine", "Delivery Truck", "Office Building", "CNC Machine", dan "Warehouse Forklift"

Metrik:

- Straight-Line (M) (skyblue): Depresiasi tahunan dengan metode garis lurus, masing-masing 50 juta, 40 juta, 50 juta, 37,5 juta, dan 16 juta.
- Double-Declining (M) (lightblue): Depresiasi dengan metode saldo menurun, masing-masing 100 juta, 80 juta, 100 juta, 75 juta, dan 32 juta.
- Tax Savings (SL, M) (lightcyan): Penghematan pajak dari metode garis lurus (25% dari depresiasi), masing-masing 12,5 juta, 10 juta, 12,5 juta, 9,375 juta, dan 4 juta.

Simulasi ini membandingkan dua metode depresiasi, menunjukkan bahwa metode saldo menurun menghasilkan depresiasi lebih tinggi di awal, yang dapat menghemat pajak lebih besar pada periode awal.

c. Normal Simulation

Subplot ini menggambarkan dua kategori transaksi: "Revenue" dan "Expense".

Metrik:

- Amount (M) (lightgreen): Total pendapatan 48 juta dan pengeluaran -7 juta.
- Net Before Tax (M) (limegreen): Laba sebelum pajak sebesar 41 juta (hanya untuk Revenue).
- Net After Tax (M) (forestgreen): Laba setelah pajak sebesar 29 juta (setelah pajak 25%, tanpa insentif ekspor).

Simulasi ini memberikan gambaran dasar kinerja keuangan, menunjukkan laba bersih yang signifikan, tetapi pajak mengurangi arus kas yang tersedia.

2.3. Kode

```
# Install Seaborn jika belum ada
!pip install seaborn

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

```

import seaborn as sns

# Mengatur tema menggunakan Seaborn
sns.set_style('darkgrid')

# Data dari simulasi sebelumnya
# 1. Tax Holiday Simulation: Pajak dan penghematan
tax_holiday_data = {
    'Transaction': ['Sale of golf clubs', 'Sale of golf bags',
'Export golf equipment'],
    'Amount': [15000000, 8000000, 25000000],
    'Tax Amount': [3750000, 2000000, 0], # 25% untuk domestik, 0%
untuk ekspor
    'Tax Savings': [0, 0, 6250000] # Penghematan dari insentif
ekspor (25% dari 25 juta)
}

# 2. Depreciation Simulation: Depresiasi tahunan (Straight-Line dan
Double-Declining)
depreciation_data = {
    'Asset': ['Injection Molding Machine', 'Delivery Truck', 'Office
Building', 'CNC Machine', 'Warehouse Forklift'],
    'Straight-Line': [50000000, 40000000, 50000000, 37500000,
16000000],
    'Double-Declining': [100000000, 80000000, 100000000, 75000000,
32000000],
    'Tax Savings (SL)': [12500000, 10000000, 12500000, 9375000,
4000000] # 25% dari Straight-Line
}

# 3. Normal Simulation: Pendapatan dan Pengeluaran
normal_data = {
    'Category': ['Revenue', 'Expense'],
    'Amount': [48000000, -7000000],
    'Net Before Tax': [41000000, 0], # Revenue - Expense
    'Net After Tax': [29000000, 0] # Setelah pajak 25% (tanpa
insentif ekspor)
}

# Membuat DataFrame untuk setiap simulasi
df_tax_holiday = pd.DataFrame(tax_holiday_data)
df_depreciation = pd.DataFrame(depreciation_data)
df_normal = pd.DataFrame(normal_data)

# Membuat grafik 3D-like untuk presentasi
fig = plt.figure(figsize=(18, 12))

# 1. Tax Holiday Simulation (Bar 3D)
ax1 = fig.add_subplot(131, projection='3d')
x1 = np.arange(len(df_tax_holiday['Transaction']))
y1 = np.array([1, 2, 3]) # Untuk memberikan efek 3D
z1 = np.zeros(len(x1))
dx1 = np.ones(len(x1)) * 0.4
dy1 = np.ones(len(x1)) * 0.4
dz1_amount = df_tax_holiday['Amount'] / 1e6 # Skala ke juta untuk
visualisasi

```

```

dz1_tax = df_tax_holiday['Tax Amount'] / 1e6
dz1_savings = df_tax_holiday['Tax Savings'] / 1e6

# Plot 3D bars dengan efek shadow
ax1.bar3d(x1, y1-0.5, z1, dx1, dy1, dz1_amount, color='salmon',
shade=True, alpha=0.8, label='Amount (M)')
ax1.bar3d(x1, y1, z1, dx1, dy1, dz1_tax, color='coral', shade=True,
alpha=0.8, label='Tax Amount (M)')
ax1.bar3d(x1, y1+0.5, z1, dx1, dy1, dz1_savings, color='lightcoral',
shade=True, alpha=0.8, label='Tax Savings (M)')
ax1.set_xticks(x1)
ax1.set_xticklabels(df_tax_holiday['Transaction'], rotation=45,
ha='right')
ax1.set_yticks([1, 2, 3])
ax1.set_yticklabels(['Amount', 'Tax', 'Savings'])
ax1.set_zlabel('Value (in Millions IDR)')
ax1.set_title('Tax Holiday Simulation', fontsize=12, pad=20)
ax1.legend()

# 2. Depreciation Simulation (Bar 3D)
ax2 = fig.add_subplot(132, projection='3d')
x2 = np.arange(len(df_depreciation['Asset']))
y2 = np.array([1, 2, 3]) # Sesuaikan dengan jumlah metrik
z2 = np.zeros(len(x2))
dx2 = np.ones(len(x2)) * 0.4
dy2 = np.ones(len(x2)) * 0.4
dz2_sl = df_depreciation['Straight-Line'] / 1e6
dz2_dd = df_depreciation['Double-Declining'] / 1e6
dz2_tax_savings = df_depreciation['Tax Savings (SL)'] / 1e6

# Pastikan semua array memiliki panjang yang sama
for i in range(len(x2)):
    ax2.bar3d(x2[i], y2[0], z2[i], dx2[i], dy2[i], dz2_sl[i],
color='skyblue', shade=True, alpha=0.8, label='Straight-Line (M)' if
i == 0 else '')
    ax2.bar3d(x2[i], y2[1], z2[i], dx2[i], dy2[i], dz2_dd[i],
color='lightblue', shade=True, alpha=0.8, label='Double-Declining
(M)' if i == 0 else '')
    ax2.bar3d(x2[i], y2[2], z2[i], dx2[i], dy2[i],
dz2_tax_savings[i], color='lightcyan', shade=True, alpha=0.8,
label='Tax Savings (SL, M)' if i == 0 else '')

ax2.set_xticks(x2)
ax2.set_xticklabels(df_depreciation['Asset'], rotation=45,
ha='right')
ax2.set_yticks([1, 2, 3])
ax2.set_yticklabels(['Straight-Line', 'Double-Declining', 'Tax
Savings'])
ax2.set_zlabel('Value (in Millions IDR)')
ax2.set_title('Depreciation Simulation', fontsize=12, pad=20)
ax2.legend()

# 3. Normal Simulation (Bar 3D)
ax3 = fig.add_subplot(133, projection='3d')
x3 = np.arange(len(df_normal['Category']))
y3 = np.array([1, 2, 3])

```

```

z3 = np.zeros(len(x3))
dx3 = np.ones(len(x3)) * 0.4
dy3 = np.ones(len(x3)) * 0.4
dz3_amount = df_normal['Amount'] / 1e6
dz3_net_before = df_normal['Net Before Tax'] / 1e6
dz3_net_after = df_normal['Net After Tax'] / 1e6

for i in range(len(x3)):
    ax3.bar3d(x3[i], y3[0], z3[i], dx3[i], dy3[i], dz3_amount[i],
color='lightgreen', shade=True, alpha=0.8, label='Amount (M)' if i
== 0 else '')
    ax3.bar3d(x3[i], y3[1], z3[i], dx3[i], dy3[i], dz3_net_before[i],
color='limegreen', shade=True, alpha=0.8, label='Net Before Tax (M)'
if i == 0 else '')
    ax3.bar3d(x3[i], y3[2], z3[i], dx3[i], dy3[i], dz3_net_after[i],
color='forestgreen', shade=True, alpha=0.8, label='Net After Tax
(M)' if i == 0 else '')

ax3.set_xticks(x3)
ax3.set_xticklabels(df_normal['Category'], rotation=45, ha='right')
ax3.set_yticks([1, 2, 3])
ax3.set_yticklabels(['Amount', 'Net Before Tax', 'Net After Tax'])
ax3.set_zlabel('Value (in Millions IDR)')
ax3.set_title('Normal Simulation', fontsize=12, pad=20)
ax3.legend()

# Menyesuaikan layout agar lebih rapi
plt.suptitle('Impact of Simulations on GolfHouse Financials',
fontsize=16, y=1.05)
plt.tight_layout()
plt.show()

```

BAB III. PENUTUP

3.1. Kesimpulan

Untuk mendukung analisis keuangan dan pengambilan keputusan strategis, GolfHouse dapat memanfaatkan Google BigQuery untuk berbagai kebutuhan pemrosesan data yang intensif. BigQuery sangat cocok untuk menangani pemrosesan transaksi keuangan dalam skala besar, menghasilkan laporan rutin seperti total pendapatan, depresiasi, atau pajak, serta mendukung integrasi dengan alat Business Intelligence (BI) seperti Looker Studio untuk visualisasi dan pelaporan yang lebih interaktif. Di sisi lain, Python di Google Colab menawarkan fleksibilitas untuk analisis eksplorasi, misalnya membandingkan metode depresiasi secara visual melalui grafik, simulasi skenario kustom seperti dampak tax holiday terhadap laba, dan prototipe model prediktif jika data diperluas di masa mendatang. Untuk hasil optimal, kombinasi kedua alat ini dapat diterapkan: BigQuery digunakan untuk pemrosesan data awal seperti agregasi dan pembersihan, kemudian hasilnya diekspor ke Colab untuk analisis lanjutan atau visualisasi menggunakan library seperti Pandas dan Matplotlib. Sebagai contoh, depresiasi dapat dihitung di BigQuery dan divisualisasikan dalam bentuk grafik di Colab, memastikan analisis yang komprehensif dan presentasi yang menarik bagi manajemen.

Penggunaan BigQuery dan Python di Colab secara terpadu memberikan pendekatan yang efisien dan fleksibel bagi GolfHouse dalam mengelola dan menganalisis data keuangan. BigQuery menjamin performa tinggi untuk data skala besar dan laporan rutin, sementara Colab memungkinkan eksplorasi mendalam dan visualisasi yang inovatif. Kombinasi ini tidak hanya meningkatkan akurasi dan kecepatan analisis, tetapi juga mendukung pengambilan keputusan berbasis data yang strategis, terutama dengan memanfaatkan insentif pajak, manajemen depresiasi, dan pemantauan laba kotor secara real-time.