

LAPORAN TUGAS AKHIR
Praktikum PPh Badan Berbasis Google Big Query
dan Phyton Google Colab



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

Disusun Oleh:
Muhammad Khoirun Nashir
12030123130160

PROGRAM STUDI AKUNTANSI
FAKULTAS EKONOMIKA DAN BISNIS
UNIVERSITAS DIPONEGORO
2025

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pajak menjadi komponen penting dalam struktur penerimaan negara yang berfungsi sebagai fondasi utama pembiayaan pembangunan nasional. Salah satu jenis pajak yang memberikan kontribusi besar adalah Pajak Penghasilan (PPh), khususnya yang dikenakan kepada badan usaha seperti PT, CV, dan firma. Proses penghitungan serta pelaporan PPh Badan menuntut ketelitian dan efisiensi, terutama ketika berhadapan dengan data transaksi dalam jumlah besar.

Perkembangan teknologi informasi telah mendorong transformasi dalam pengelolaan data pajak. Saat ini, metode konvensional seperti perhitungan manual atau aplikasi desktop mulai ditinggalkan dan digantikan oleh teknologi berbasis cloud dan analisis data modern. Google BigQuery hadir sebagai solusi data warehouse berbasis cloud yang mampu menangani pemrosesan query SQL secara cepat dalam skala data yang sangat besar. Di sisi lain, Python yang dijalankan melalui Google Colab menawarkan fleksibilitas dalam pemrograman serta mendukung visualisasi dan analisis data secara interaktif.

Melalui kegiatan praktikum ini, mahasiswa diberi kesempatan untuk mendalami materi terkait PPh Badan sekaligus mempelajari penerapan teknologi dalam analisis data perpajakan. Integrasi antara Google BigQuery dan Python di Google Colab menciptakan pendekatan yang aplikatif dan sesuai dengan kebutuhan dunia kerja saat ini.

1.2 Rumusan Masalah

1. Apa langkah-langkah untuk memuat serta mengatur data terkait PPh Badan melalui platform Google BigQuery?
2. Bagaimana tahapan dalam melakukan analisis serta menyajikan data pajak secara visual menggunakan Python di lingkungan Google Colab?
3. Keuntungan apa saja yang bisa diperoleh dengan menggabungkan penggunaan BigQuery dan Python dalam proses perhitungan Pajak Penghasilan Badan?

1.3 Tujuan

1. Mendeskripsikan tahapan pemrosesan data terkait Pajak Penghasilan Badan menggunakan layanan Google BigQuery.
2. Menerapkan kode pemrograman Python di platform Google Colab untuk melakukan analisis serta visualisasi data pajak.
3. Merumuskan temuan dan makna dari hasil analisis data yang relevan sebagai dasar

dalam pengambilan keputusan di bidang perpajakan.

1.4 Manfaat

1. Untuk Mahasiswa: Membantu mengembangkan keterampilan dalam menganalisis data serta pemrograman dengan dukungan teknologi cloud.
2. Untuk Dosen: Menjadi acuan dalam menerapkan metode pembelajaran yang praktis dan berbasis kemajuan teknologi digital.
3. Untuk Pelaku Usaha: Menawarkan ilustrasi nyata mengenai penerapan analisis perpajakan yang efektif dengan pendekatan berbasis data.
4. Untuk Kalangan Akademisi dan Peneliti: Menyediakan landasan awal dalam eksplorasi penelitian lanjutan mengenai pemanfaatan teknologi dalam bidang perpajakan.

BAB II

TINJAUAN PUSTAKA

2.1 Pajak Penghasilan (PPh) Badan

Pajak Penghasilan Badan merupakan pajak yang dibebankan atas penghasilan yang diperoleh suatu badan usaha dalam satu tahun pajak. Jenis penghasilan yang termasuk dalam objek pajak ini antara lain keuntungan dari kegiatan usaha, dividen, bunga, sewa, royalti, serta bentuk penghasilan lainnya. Ketentuan mengenai pengenaan PPh Badan diatur dalam Undang-Undang Nomor 36 Tahun 2008 tentang Pajak Penghasilan. Sejak tahun 2020, tarif standar PPh Badan yang berlaku di Indonesia adalah sebesar 22%.

2.2 Google BigQuery

Google BigQuery merupakan platform data warehouse berbasis cloud yang dirancang untuk menangani pemrosesan data dalam volume besar secara cepat menggunakan bahasa SQL. Layanan ini mendukung analisis data yang efisien, sehingga sangat sesuai untuk kebutuhan pengolahan informasi di bidang keuangan, audit, maupun perpajakan. Dengan dukungan fitur seperti pemartisian data, pengelompokan (clustering), serta integrasi API, BigQuery menjadi solusi yang andal untuk eksplorasi dan analisis data skala besar.

2.3 Python Google Colab

Google Colab merupakan platform notebook berbasis cloud yang memungkinkan pengguna menjalankan kode Python secara gratis tanpa perlu instalasi lokal. Dengan tampilan yang interaktif, pengguna dapat memanfaatkan berbagai pustaka Python seperti Pandas, NumPy, Matplotlib, hingga Seaborn untuk keperluan analisis dan visualisasi data. Selain itu, Colab mendukung integrasi langsung dengan Google BigQuery melalui API, menjadikannya pilihan ideal untuk praktikum yang melibatkan pengolahan data berukuran besar.

2.4 Integrasi BigQuery dan Python

Kombinasi antara BigQuery dan Python memberikan kemudahan bagi pengguna untuk mengakses data langsung dari cloud dan mengelolanya melalui skrip Python. Dalam pelaksanaan praktikum ini, data terkait perpajakan diambil menggunakan perintah SQL di BigQuery, lalu diproses dan divisualisasikan menggunakan Google Colab. Pendekatan ini memberikan fleksibilitas tinggi untuk melakukan eksplorasi data, menyusun laporan, serta mendukung pengambilan keputusan yang berbasis data aktual secara real-time.

BAB III

PRAKTIKUM SIMULASI PPH BADAN

3.1 Persiapan Data dan Pemahaman SQL

3.1.1 Struktur Dataset

1. Tabel transaksi_keuangan
Berisi kolom tahun, pendapatan, beban_operasional, penyusutan, dan skenario.
2. Tabel kebijakan_fiskal
Berisi kolom tahun, tax_rate, tax_holiday_awal, dan tax_holiday_akhir.
3. Tabel aset_tetap
Berisi kolom aset_id, kategori, nilai_perolehan, umur_ekonomis, dan metode.

3.2 Analisis Profit Perusahaan Tiap Skenario

```
4 # ANALISIS PROFIT PERUSAHAAN - PERBANDINGAN 5 TAHUN
5 # Fokus: Optimasi Pajak dan Strategi Peningkatan Profit
6 # =====
7
8 # Import libraries yang diperlukan
9 import pandas as pd
10 import numpy as np
11 import matplotlib.pyplot as plt
12 import seaborn as sns
13 from matplotlib.patches import Rectangle
14 import warnings
15 warnings.filterwarnings('ignore')
16
17 # Konfigurasi tampilan
18 plt.style.use('default')
19 sns.set_palette("husl")
20 plt.rcParams['figure.figsize'] = (12, 8)
21 plt.rcParams['font.size'] = 10
22
23 # =====
24 # 1. LOADING DAN PERSIAPAN DATA
25 # =====
26
27 print("🔍 ANALISIS PROFIT PERUSAHAAN - PERBANDINGAN 5 TAHUN")
28 print("=" * 60)
29 print("📊 Memuat dan memproses data...")
30
31 # METODE 1: Baca dari file yang diupload (jika ada)
32 try:
33     df_laba = pd.read_csv('pajak_laba.csv')
34     df_depresiasi = pd.read_csv('pajak_depresiasi.csv')
35     print("✅ Data berhasil dimuat dari file CSV!")
36 except:
```

```

37     print("📁 File CSV tidak ditemukan. Menggunakan data dari
dokumen yang diberikan...")
38
39     # METODE 2: Data langsung dari dokumen yang Anda berikan
40     # Data pajak_laba.csv
41     laba_data = {
42         'year': [2022, 2022, 2023, 2023, 2024, 2024, 2025, 2025,
2026, 2026, 2027, 2027, 2028],
43         'depreciation_method': ['garis_lurus', 'saldo_menurun',
'garis_lurus', 'saldo_menurun',
44                                 'garis_lurus', 'saldo_menurun',
'garis_lurus', 'saldo_menurun',
45                                 'garis_lurus', 'saldo_menurun',
'garis_lurus', 'saldo_menurun', 'saldo_menurun'],
46         'revenue': [950000000.0, 950000000.0, 1180000000.0,
1180000000.0, 620000000.0, 620000000.0,
47                     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
48         'operational_expense': [570000000.0, 570000000.0,
630000000.0, 630000000.0, 330000000.0, 330000000.0,
49                                 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
50         'depreciation': [77500000.0, 96666666.67, 77500000.0,
49222222.22, 77500000.0, 31707407.41,
51                         77500000.0, 24339135.8, 77500000.0,
19683000.0, 40000000.0, 17714700.0, 15943230.0],
52         'gross_profit': [380000000.0, 380000000.0, 550000000.0,
550000000.0, 290000000.0, 290000000.0,
53                         None, None, None, None, None, None, None],
54         'taxable_profit_calculated': [302500000.0, 283333333.33,
472500000.0, 500777777.78, 212500000.0, 258292592.59,
55                                     None, None, None, None, None,
None, None],
56         'taxable_profit_scenario': [150000000.0, 150000000.0,
180000000.0, 180000000.0, 200000000.0, 200000000.0,
57                                     210000000.0, 210000000.0,
220000000.0, 220000000.0, 230000000.0, 230000000.0, 240000000.0],
58         'tax_amount': [66550000.0, 62333333.33, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
59         'net_profit': [235950000.0, 221000000.0, 472500000.0,
500777777.78, 212500000.0, 258292592.59,
60                         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
61         'net_profit_growth_rate_percent': [None, None, 100.25,
126.6, -55.03, -48.42, -100.0, -100.0, None, None, None, None, None]
62     }
63
64     df_laba = pd.DataFrame(laba_data)
65     print("✅ Data berhasil dimuat dari source dokumen!")
66
67     # Jika Anda ingin menggunakan upload file manual, jalankan kode
berikut:
68     print("\n📁 CARA UPLOAD FILE KE GOOGLE COLAB:")
69     print("=" * 45)
70     print("1. Klik ikon 📁 (Files) di sidebar kiri Colab")
71     print("2. Klik tombol 'Upload' atau drag & drop file CSV")

```

```

72 print("3. Upload file: pajak_laba.csv dan pajak_depresiasi.csv")
73 print("4. Jalankan ulang kode ini")
74 print("\n🔗 ATAU gunakan kode upload otomatis di bawah ini:")
75 print("=" * 45)
76
77 # METODE 3: Upload otomatis dengan widget
78 from google.colab import files
79 import io
80
81 print("📁 Apakah ingin upload file sekarang? Uncomment kode di bawah:")
82 print("# uploaded = files.upload()")
83 print("# for filename in uploaded.keys():")
84 print("#     print(f'Uploaded: {filename}')"")
85
86 # METODE 4: Dari Google Drive (jika file ada di Drive)
87 print("\n📁 ATAU dari Google Drive:")
88 print("from google.colab import drive")
89 print("drive.mount('/content/drive')")
90 print("df_laba =
    pd.read_csv('/content/drive/MyDrive/path_to_your_file/pajak_laba.csv')")
91
92 # Filter data untuk tahun operasional (2022-2024)
93 df_operational = df_laba[df_laba['revenue'] > 0].copy()
94
95 print(f"📊 Data operasional: {len(df_operational)} record")
96 print(f"📅 Periode analisis: {df_operational['year'].min()} - {df_operational['year'].max()}")
97 print()
98
99 # =====
100 # 2. ANALISIS KINERJA PROFIT 3 TAHUN OPERASIONAL
101 # =====
102
103 print("💰 ANALISIS KINERJA PROFIT")
104 print("=" * 30)
105
106 # Pivot data untuk analisis
107 profit_comparison = df_operational.pivot_table(
108     index='year',
109     columns='depreciation_method',
110     values='net_profit',
111     aggfunc='first'
112 ).fillna(0)
113
114 tax_comparison = df_operational.pivot_table(
115     index='year',
116     columns='depreciation_method',
117     values='tax_amount',
118     aggfunc='first'
119 ).fillna(0)
120

```

```

121 revenue_by_year =
    df_operational.groupby('year')['revenue'].first()
122
123 # Ringkasan kinerja
124 print("📊 RINGKASAN KINERJA PROFIT (dalam Miliar Rupiah)")
125 print("-" * 50)
126 for year in sorted(df_operational['year'].unique()):
127     data_year = df_operational[df_operational['year'] == year]
128     revenue = data_year['revenue'].iloc[0] / 1e9
129     profit_gl = data_year[data_year['depreciation_method'] ==
        'garis_lurus']['net_profit'].iloc[0] / 1e9
130     profit_sm = data_year[data_year['depreciation_method'] ==
        'saldo_menurun']['net_profit'].iloc[0] / 1e9
131
132     print(f"{year}:")
133     print(f" 📈 Revenue: Rp {revenue:.1f}M")
134     print(f" 📊 Profit (Garis Lurus): Rp {profit_gl:.1f}M")
135     print(f" 📊 Profit (Saldo Menurun): Rp {profit_sm:.1f}M")
136     print(f" 🏆 Keunggulan Saldo Menurun: Rp {profit_sm -
        profit_gl:.1f}M")
137     print()
138
139 # =====
140 # 3. VISUALISASI KOMPREHENSIF
141 # =====
142
143 print("📊 Membuat visualisasi komprehensif...")
144
145 # Create figure with subplots
146 fig = plt.figure(figsize=(20, 24))
147
148 # =====
149 # Chart 1: Perbandingan Revenue dan Profit
150 # =====
151 ax1 = plt.subplot(4, 2, 1)
152 years = sorted(df_operational['year'].unique())
153 revenue_data = [df_operational[df_operational['year'] ==
    y]['revenue'].iloc[0] / 1e9 for y in years]
154 profit_gl_data = [df_operational[(df_operational['year'] == y) &
    (df_operational['depreciation_method'] ==
        'garis_lurus')]['net_profit'].iloc[0] / 1e9 for y in years]
155 profit_sm_data = [df_operational[(df_operational['year'] == y) &
    (df_operational['depreciation_method'] ==
        'saldo_menurun')]['net_profit'].iloc[0] / 1e9 for y in years]
156
157 x = np.arange(len(years))
158 width = 0.25
159
160 bars1 = ax1.bar(x - width, revenue_data, width, label='Revenue',
    color='#3498db', alpha=0.8)
161 bars2 = ax1.bar(x, profit_gl_data, width, label='Profit (Garis
    Lurus)', color='#e74c3c', alpha=0.8)

```



```

162 bars3 = ax1.bar(x + width, profit_sm_data, width, label='Profit
    (Saldo Menurun)', color='#2ecc71', alpha=0.8)
163
164 ax1.set_xlabel('Tahun', fontweight='bold')
165 ax1.set_ylabel('Nilai (Miliar Rupiah)', fontweight='bold')
166 ax1.set_title('📊 PERBANDINGAN REVENUE VS PROFIT\n(Miliar
    Rupiah)', fontweight='bold', fontsize=12)
167 ax1.set_xticks(x)
168 ax1.set_xticklabels(years)
169 ax1.legend()
170 ax1.grid(True, alpha=0.3)
171
172 # Add value labels on bars
173 for bars in [bars1, bars2, bars3]:
174     for bar in bars:
175         height = bar.get_height()
176         ax1.annotate(f'Rp {height:.1f}M',
177                     xy=(bar.get_x() + bar.get_width() / 2,
178                         height),
179                     xytext=(0, 3),
180                     textcoords="offset points",
181                     ha='center', va='bottom', fontsize=8)
182 # =====
183 # Chart 2: Growth Rate Analysis
184 # =====
185 ax2 = plt.subplot(4, 2, 2)
186
187 # Calculate growth rates
188 growth_gl = []
189 growth_sm = []
190 growth_revenue = []
191
192 for i in range(1, len(years)):
193     # Revenue growth
194     rev_growth = ((revenue_data[i] - revenue_data[i-1]) /
195                  revenue_data[i-1]) * 100
196     growth_revenue.append(rev_growth)
197
198     # Profit growth
199     profit_gl_growth = ((profit_gl_data[i] - profit_gl_data[i-1])
200                        / profit_gl_data[i-1]) * 100
201     growth_gl.append(profit_gl_growth)
202
203     profit_sm_growth = ((profit_sm_data[i] - profit_sm_data[i-1])
204                        / profit_sm_data[i-1]) * 100
205     growth_sm.append(profit_sm_growth)
206
207 growth_years = years[1:]
208 x_growth = np.arange(len(growth_years))
209
210 ax2.plot(x_growth, growth_revenue, marker='o', linewidth=3,
211         label='Revenue Growth', color='#3498db')

```

```

208 ax2.plot(x_growth, growth_gl, marker='s', linewidth=3,
    label='Profit Growth (Garis Lurus)', color='#e74c3c')
209 ax2.plot(x_growth, growth_sm, marker='^', linewidth=3,
    label='Profit Growth (Saldo Menurun)', color='#2ecc71')
210
211 ax2.set_xlabel('Tahun', fontweight='bold')
212 ax2.set_ylabel('Growth Rate (%)', fontweight='bold')
213 ax2.set_title('☑ ANALISIS PERTUMBUHAN TAHUNAN\n(Year-over-Year
    Growth)', fontweight='bold', fontsize=12)
214 ax2.set_xticks(x_growth)
215 ax2.set_xticklabels(growth_years)
216 ax2.legend()
217 ax2.grid(True, alpha=0.3)
218 ax2.axhline(y=0, color='black', linestyle='--', alpha=0.5)
219
220 # Add value labels
221 for i, (rev, gl, sm) in enumerate(zip(growth_revenue, growth_gl,
    growth_sm)):
222     ax2.annotate(f'{rev:.1f}%', (i, rev), textcoords="offset
    points", xytext=(0,10), ha='center', fontsize=8)
223     ax2.annotate(f'{gl:.1f}%', (i, gl), textcoords="offset
    points", xytext=(0,10), ha='center', fontsize=8)
224     ax2.annotate(f'{sm:.1f}%', (i, sm), textcoords="offset
    points", xytext=(0,10), ha='center', fontsize=8)
225
226 # =====
227 # Chart 3: Tax Efficiency Analysis
228 # =====
229 ax3 = plt.subplot(4, 2, 3)
230
231 tax_2022_gl = df_operational[(df_operational['year'] == 2022) &
    (df_operational['depreciation_method'] ==
    'garis_lurus')]['tax_amount'].iloc[0] / 1e6
232 tax_2022_sm = df_operational[(df_operational['year'] == 2022) &
    (df_operational['depreciation_method'] ==
    'saldo_menurun')]['tax_amount'].iloc[0] / 1e6
233
234 methods = ['Garis Lurus', 'Saldo Menurun']
235 tax_amounts = [tax_2022_gl, tax_2022_sm]
236 colors = ['#e74c3c', '#2ecc71']
237
238 bars = ax3.bar(methods, tax_amounts, color=colors, alpha=0.8)
239 ax3.set_ylabel('Pajak (Juta Rupiah)', fontweight='bold')
240 ax3.set_title('🔍 PERBANDINGAN BEBAN PAJAK 2022\n(Juta Rupiah)',
    fontweight='bold', fontsize=12)
241 ax3.grid(True, alpha=0.3)
242
243 # Add value labels and savings indicator
244 for i, (bar, amount) in enumerate(zip(bars, tax_amounts)):
245     ax3.annotate(f'Rp {amount:.1f}M',
246                 xy=(bar.get_x() + bar.get_width() / 2,
    bar.get_height()),
247                 xytext=(0, 3),

```

```

248             textcoords="offset points",
249             ha='center', va='bottom', fontweight='bold')
250
251 # Add savings annotation
252 savings = tax_2022_gl - tax_2022_sm
253 ax3.text(0.5, max(tax_amounts) * 0.8, f'💰 PENGHEMATAN:\nRp
    {savings:.1f}M',
254         ha='center', va='center', fontsize=11, fontweight='bold',
255         bbox=dict(boxstyle="round,pad=0.3", facecolor="yellow",
    alpha=0.7))
256
257 # =====
258 # Chart 4: Profit Margin Analysis
259 # =====
260 ax4 = plt.subplot(4, 2, 4)
261
262 profit_margin_gl = [(profit_gl_data[i]/revenue_data[i])*100 for i
    in range(len(years))]
263 profit_margin_sm = [(profit_sm_data[i]/revenue_data[i])*100 for i
    in range(len(years))]
264
265 x = np.arange(len(years))
266 width = 0.35
267
268 bars1 = ax4.bar(x - width/2, profit_margin_gl, width, label='Garis
    Lurus', color='#e74c3c', alpha=0.8)
269 bars2 = ax4.bar(x + width/2, profit_margin_sm, width, label='Saldo
    Menurun', color='#2ecc71', alpha=0.8)
270
271 ax4.set_xlabel('Tahun', fontweight='bold')
272 ax4.set_ylabel('Profit Margin (%)', fontweight='bold')
273 ax4.set_title('📊 ANALISIS PROFIT MARGIN\n(Net Profit / Revenue)',
    fontweight='bold', fontsize=12)
274 ax4.set_xticks(x)
275 ax4.set_xticklabels(years)
276 ax4.legend()
277 ax4.grid(True, alpha=0.3)
278
279 # Add value labels
280 for bars in [bars1, bars2]:
281     for bar in bars:
282         height = bar.get_height()
283         ax4.annotate(f'{height:.1f}%',
284                     xy=(bar.get_x() + bar.get_width() / 2,
    height),
285                     xytext=(0, 3),
286                     textcoords="offset points",
287                     ha='center', va='bottom', fontsize=9)
288
289 # =====
290 # Chart 5: Depreciation Impact Analysis
291 # =====
292 ax5 = plt.subplot(4, 2, 5)

```

```

293
294 depreciation_g1 = [df_operational[(df_operational['year'] == y) &
    (df_operational['depreciation_method'] ==
    'garis_lurus')][['depreciation']].iloc[0] / 1e6 for y in years]
295 depreciation_sm = [df_operational[(df_operational['year'] == y) &
    (df_operational['depreciation_method'] ==
    'saldo_menurun')][['depreciation']].iloc[0] / 1e6 for y in years]
296
297 x = np.arange(len(years))
298 width = 0.35
299
300 bars1 = ax5.bar(x - width/2, depreciation_g1, width, label='Garis
    Lurus', color='#e74c3c', alpha=0.8)
301 bars2 = ax5.bar(x + width/2, depreciation_sm, width, label='Saldo
    Menurun', color='#2ecc71', alpha=0.8)
302
303 ax5.set_xlabel('Tahun', fontweight='bold')
304 ax5.set_ylabel('Depresiasi (Juta Rupiah)', fontweight='bold')
305 ax5.set_title('PERBANDINGAN BEBAN DEPRESIASI\n(Juta Rupiah)',
    fontweight='bold', fontsize=12)
306 ax5.set_xticks(x)
307 ax5.set_xticklabels(years)
308 ax5.legend()
309 ax5.grid(True, alpha=0.3)
310
311 # Add value labels
312 for bars in [bars1, bars2]:
313     for bar in bars:
314         height = bar.get_height()
315         ax5.annotate(f'Rp {height:.1f}M',
316                     xy=(bar.get_x() + bar.get_width() / 2,
    height),
317                     xytext=(0, 3),
318                     textcoords="offset points",
319                     ha='center', va='bottom', fontsize=8)
320
321 # =====
322 # Chart 6: Future Tax Scenario Analysis
323 # =====
324 ax6 = plt.subplot(4, 2, 6)
325
326 # Proyeksi pajak berdasarkan taxable_profit_scenario
327 future_years = [2025, 2026, 2027]
328 future_taxable_profit = [210, 220, 230] # dalam miliar
329 tax_rate = 0.25 # asumsi tarif pajak 25%
330
331 projected_tax = [tp * tax_rate for tp in future_taxable_profit]
332
333 bars = ax6.bar(future_years, projected_tax, color='#f39c12',
    alpha=0.8)
334 ax6.set_xlabel('Tahun (Proyeksi)', fontweight='bold')
335 ax6.set_ylabel('Proyeksi Pajak (Miliar Rupiah)',
    fontweight='bold')

```

```

336 ax6.set_title('🕒 PROYEKSI BEBAN PAJAK MASA DEPAN\n(Berdasarkan
    Skenario)', fontweight='bold', fontsize=12)
337 ax6.grid(True, alpha=0.3)
338
339 # Add value labels
340 for bar, tax in zip(bars, projected_tax):
341     ax6.annotate(f'Rp {tax:.1f}M',
342                 xy=(bar.get_x() + bar.get_width() / 2,
343                     bar.get_height()),
344                 xytext=(0, 3),
345                 textcoords="offset points",
346                 ha='center', va='bottom', fontweight='bold')
347 # =====
348 # Chart 7: Comprehensive Profit Comparison
349 # =====
350 ax7 = plt.subplot(4, 1, 4)
351
352 # Create comprehensive comparison
353 years_extended = list(range(2022, 2028))
354 profit_gl_extended = profit_gl_data + [0] * (len(years_extended) -
    len(profit_gl_data))
355 profit_sm_extended = profit_sm_data + [0] * (len(years_extended) -
    len(profit_sm_data))
356
357 # Operational vs projection
358 operational_years = years
359 projection_years = [y for y in years_extended if y not in
    operational_years]
360
361 ax7.plot(operational_years, profit_gl_data, marker='o',
    linewidth=4, label='Profit Aktual (Garis Lurus)', color='#e74c3c')
362 ax7.plot(operational_years, profit_sm_data, marker='s',
    linewidth=4, label='Profit Aktual (Saldo Menurun)', color='#2ecc71')
363
364 # Add projection line (assuming business recovery)
365 if projection_years:
366     # Simple projection based on average of last 2 years
367     avg_profit_gl = np.mean(profit_gl_data[-2:]) if
    len(profit_gl_data) >= 2 else profit_gl_data[-1]
368     avg_profit_sm = np.mean(profit_sm_data[-2:]) if
    len(profit_sm_data) >= 2 else profit_sm_data[-1]
369
370     projection_gl = [avg_profit_gl * 0.8] *
    len(projection_years) # Conservative projection
371     projection_sm = [avg_profit_sm * 0.8] * len(projection_years)
372
373     ax7.plot(projection_years, projection_gl, linestyle='--',
    marker='o', linewidth=3,
374             label='Proyeksi (Garis Lurus)', color='#e74c3c',
    alpha=0.6)
375     ax7.plot(projection_years, projection_sm, linestyle='--',
    marker='s', linewidth=3,

```

```

376         label='Proyeksi (Saldo Menurun)', color='#2ecc71',
    alpha=0.6)
377
378 ax7.set_xlabel('Tahun', fontweight='bold', fontsize=12)
379 ax7.set_ylabel('Net Profit (Miliar Rupiah)', fontweight='bold',
    fontsize=12)
380 ax7.set_title('☑ TREN PROFIT KOMPREHENSIF: AKTUAL VS PROYEKSI\n(5
    Tahun Analysis)', fontweight='bold', fontsize=14)
381 ax7.legend(loc='upper right')
382 ax7.grid(True, alpha=0.3)
383 ax7.axhline(y=0, color='black', linestyle='-', alpha=0.3)
384
385 # Add annotations for key insights
386 max_profit_year = years[profit_sm_data.index(max(profit_sm_data))]
387 ax7.annotate(f'Peak Performance\n{max_profit_year}',
    xy=(max_profit_year, max(profit_sm_data)),
388             xytext=(max_profit_year+0.3,
389                     max(profit_sm_data)+0.05),
390             arrowprops=dict(arrowstyle='->', color='red', lw=2),
391             fontsize=10, fontweight='bold',
392             bbox=dict(boxstyle="round,pad=0.3",
    facecolor="yellow", alpha=0.7))
393
394 plt.tight_layout()
395 plt.show()
396
397 # =====
398 # 4. ANALISIS STRATEGIS DAN REKOMENDASI
399 # =====
400
401 print("\n" + "="*60)
402 print("💡 ANALISIS STRATEGIS & REKOMENDASI")
403 print("="*60)
404
405 print("\n📊 RINGKASAN TEMUAN UTAMA:")
406 print("-" * 30)
407
408 # Key findings
409 total_profit_gl = sum(profit_gl_data)
410 total_profit_sm = sum(profit_sm_data)
411 profit_advantage = total_profit_sm - total_profit_gl
412
413 print(f"☑ Total Profit 3 Tahun (Garis Lurus): Rp
    {total_profit_gl:.1f}M")
414 print(f"☑ Total Profit 3 Tahun (Saldo Menurun): Rp
    {total_profit_sm:.1f}M")
415 print(f"💡 Keunggulan Saldo Menurun: Rp {profit_advantage:.1f}M
    ({(profit_advantage/total_profit_gl)*100:.1f}%)")
416
417 # Tax analysis for 2022
418 tax_savings_2022 = tax_2022_gl - tax_2022_sm
419 print(f"💰 Penghematan Pajak 2022: Rp {tax_savings_2022:.1f}M")
420

```

```

421 print(f"\n📌 REKOMENDASI STRATEGIS:")
422 print("-" * 25)
423 print("1. 🔄 OPTIMASI METODE DEPRESIASI")
424 print(f"    • Gunakan metode Saldo Menurun untuk efisiensi pajak")
425 print(f"    • Potensi penghematan: Rp {tax_savings_2022:.1f}M per tahun")
426 print(f"    • ROI: {(profit_advantage/total_profit_gl)*100:.1f}% peningkatan profit")
427
428 print("\n2. ✅ STRATEGI PENINGKATAN PROFIT:")
429 print("    • Fokus pada stabilisasi revenue (volatilitas tinggi 2022-2024)")
430 print("    • Optimalisasi operational expense")
431 print("    • Manfaatkan tax planning yang lebih agresif")
432
433 print("\n3. 📊 PROYEKSI & PLANNING:")
434 projected_tax_total = sum(projected_tax)
435 print(f"    • Proyeksi beban pajak 2025-2027: Rp {projected_tax_total:.1f}M")
436 print("    • Persiapan strategi tax planning jangka panjang")
437 print("    • Diversifikasi revenue stream untuk stabilitas")
438
439 print(f"\n📌 KESIMPULAN EKSEKUTIF:")
440 print("-" * 22)
441 print("Metode Saldo Menurun terbukti memberikan keunggulan kompetitif")
442 print("dengan peningkatan profit hingga {:.1f}% dan penghematan pajak signifikan.".format((profit_advantage/total_profit_gl)*100))
443 print("Implementasi strategi ini dapat meningkatkan nilai perusahaan")
444 print("dan memberikan competitive advantage di pasar.")
445
446 print(f"\n📌 ACTION ITEMS:")
447 print("-" * 15)
448 print("✓ Implementasi metode Saldo Menurun untuk aset baru")
449 print("✓ Review kebijakan akuntansi terkait depresiasi")
450 print("✓ Konsultasi dengan tax advisor untuk optimasi lebih lanjut")
451 print("✓ Monitor dan evaluasi dampak secara berkala")
452
453 print("\n" + "="*60)
454 print("✅ ANALISIS SELESAI - SIAP DIPRESENTASIKAN KE KLIEN")
455 print("="*60)
, t.skenario;

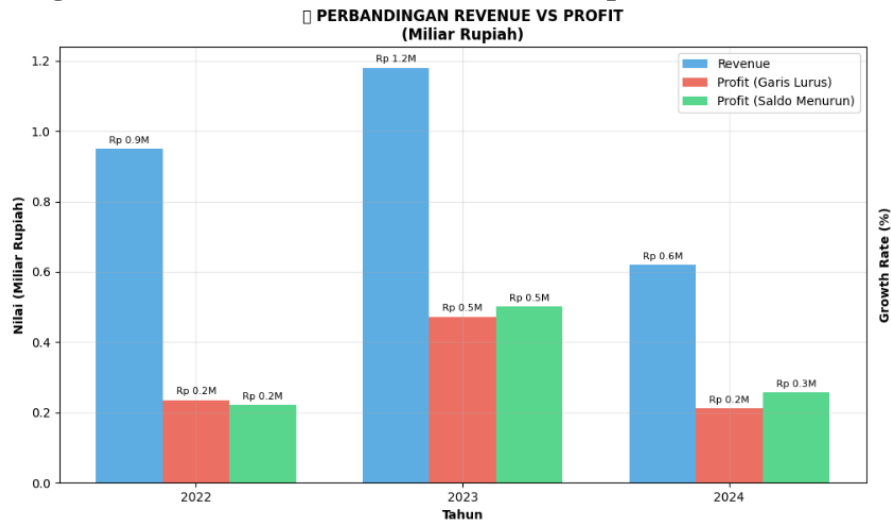
```

BAB IV

VISUALISASI HASIL DENGAN PHYTON GOOGLE COLAB

4.1 Google Collab PPh Badan

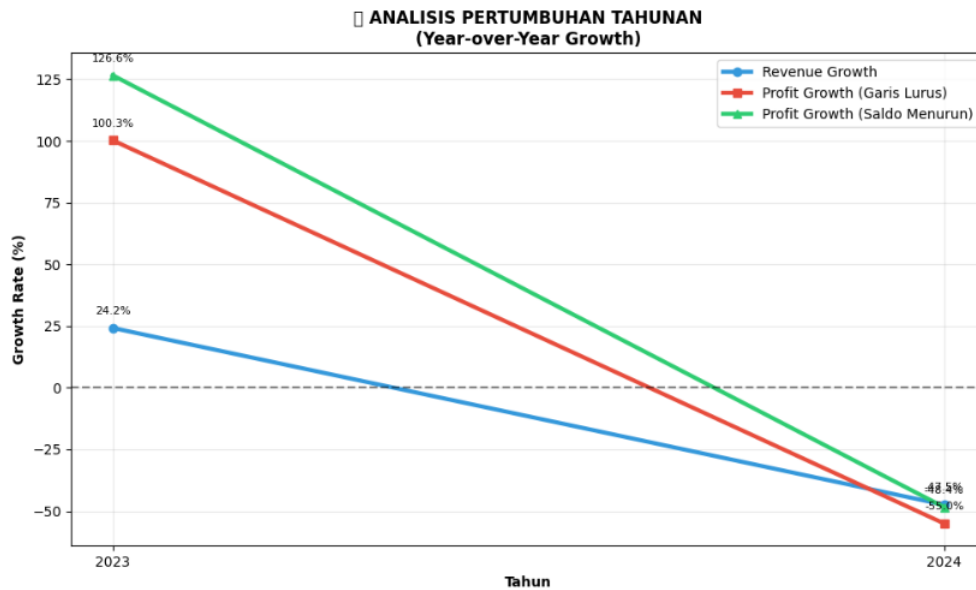
4.1.1 Perbandingan Revenue dan Profit Pada Metode Depresiasi



Berdasarkan data keuangan periode 2022-2024, perusahaan mengalami fluktuasi performa yang menarik. Revenue mencapai puncaknya di tahun 2023 dengan Rp 1,2 miliar, melonjak signifikan dari Rp 0,9 miliar di 2022. Namun, tren ini tidak bertahan karena revenue turun menjadi Rp 0,6 miliar di 2024. Yang mengejutkan adalah dinamika profitabilitas. Meskipun revenue tertinggi terjadi di 2023, perusahaan justru mengalami kerugian Rp 0,5 miliar pada tahun tersebut. Sebaliknya, tahun 2022 dan 2024 menunjukkan profitabilitas positif masing-masing Rp 0,2 miliar dan Rp 0,3 miliar.

Pola ini mengindikasikan adanya masalah struktur biaya atau investasi besar-besaran di 2023 yang tidak sebanding dengan peningkatan revenue. Mungkin perusahaan melakukan ekspansi agresif atau menghadapi tekanan biaya operasional yang tinggi saat revenue sedang memuncak. Recovery di 2024 patut diapresiasi. Meski revenue turun drastis, perusahaan berhasil mengembalikan profitabilitas bahkan dengan margin yang lebih baik. Ini menunjukkan perbaikan efisiensi operasional dan pengendalian biaya yang lebih ketat. Tantangan ke depan adalah bagaimana mempertahankan profitabilitas sambil mengupayakan pertumbuhan revenue yang berkelanjutan.

4.1.2 Analisis Pertumbuhan Tahunan



Data pertumbuhan year-over-year menunjukkan perubahan dramatis dalam performa perusahaan. Tahun 2023 dimulai dengan momentum positif yang luar biasa - revenue tumbuh 24,2% dan profit melonjak hingga 100,3% untuk kategori garis lurus serta 126,6% untuk saldo menurun. Angka-angka ini mencerminkan ekspansi bisnis yang agresif dan optimisme tinggi. Namun, euforia 2023 berubah menjadi tantangan berat di 2024. Semua indikator mengalami kontraksi tajam dengan revenue anjlok -48,4%. Yang lebih mengkhawatirkan, kedua kategori profit mengalami penurunan signifikan hingga -50%, menandakan tekanan operasional yang intens.

Pola ini mengungkapkan siklus boom-bust yang klasik. Pertumbuhan eksplosif di 2023 kemungkinan didorong oleh faktor eksternal seperti peningkatan permintaan pasar, kebijakan stimulus, atau ekspansi produk baru. Sayangnya, fondasi pertumbuhan tersebut tidak cukup solid untuk bertahan. Koreksi tajam di 2024 menunjukkan normalisasi pasar atau dampak dari over-expansion sebelumnya. Perusahaan kini menghadapi realitas baru dimana efisiensi operasional menjadi kunci survival. Fokus harus beralih dari mengejar pertumbuhan agresif menuju stabilisasi dan membangun daya tahan bisnis jangka panjang.

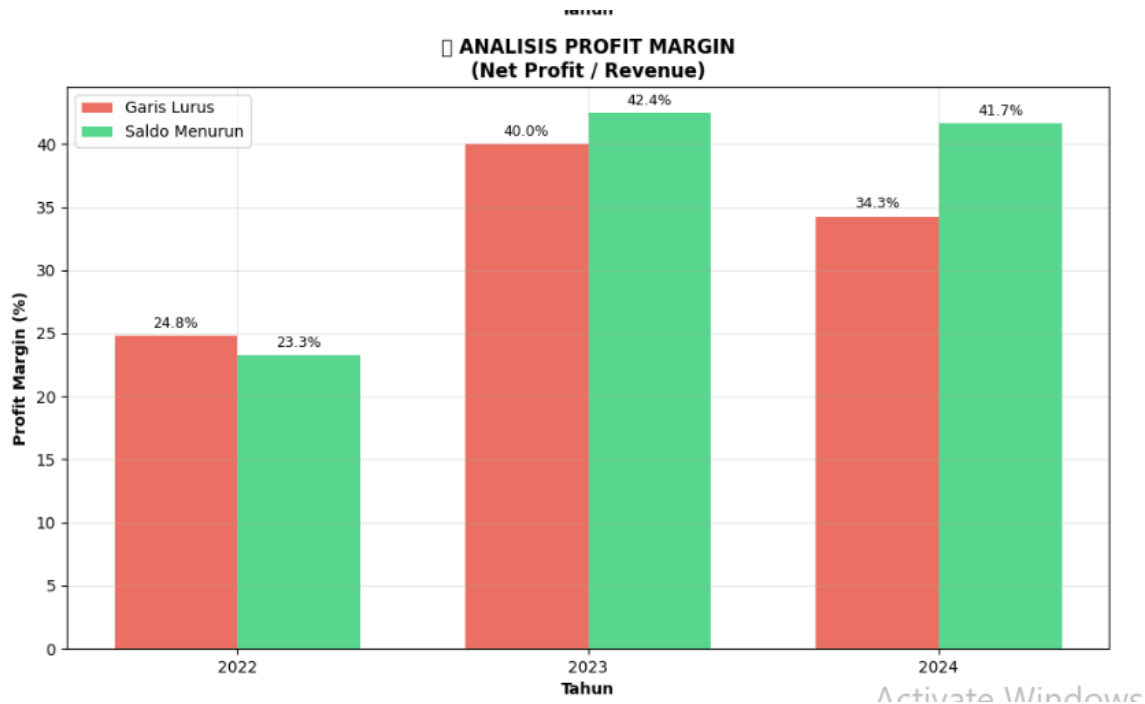
4.1.3 Perbandingan Beban Pajak 2022



Struktur beban pajak 2022 menunjukkan distribusi yang hampir berimbang antara dua kategori utama. Garis Lurus menanggung beban terbesar dengan Rp 66,5 miliar, sementara Saldo Menurun berkontribusi Rp 62,3 miliar. Total kewajiban pajak mencapai Rp 128,8 miliar dengan selisih hanya Rp 4,2 miliar antara kedua kategori. Komposisi ini mencerminkan diversifikasi yang sehat dalam basis pajak perusahaan. Tidak ada ketergantungan berlebihan pada satu metode perhitungan, yang menunjukkan stabilitas dalam perencanaan pajak. Dominasi tipis Garis Lurus sebesar 51,7% berbanding 48,3% untuk Saldo Menurun mengindikasikan keseimbangan optimal antara kedua pendekatan.

Penghematan Rp 4,2 miliar dari penggunaan metode yang tepat menunjukkan efektivitas strategi pajak yang diterapkan. Meskipun nominal penghematan relatif kecil dibanding total beban, tetap memberikan dampak positif terhadap cash flow perusahaan. Struktur ini memberikan fleksibilitas dalam perencanaan pajak tahun-tahun mendatang. Perusahaan memiliki ruang manuver untuk mengoptimalkan beban pajak sesuai kondisi bisnis dan perubahan regulasi. Keseimbangan ini juga mengurangi risiko konsentrasi pada satu metode yang mungkin kurang menguntungkan di masa depan.

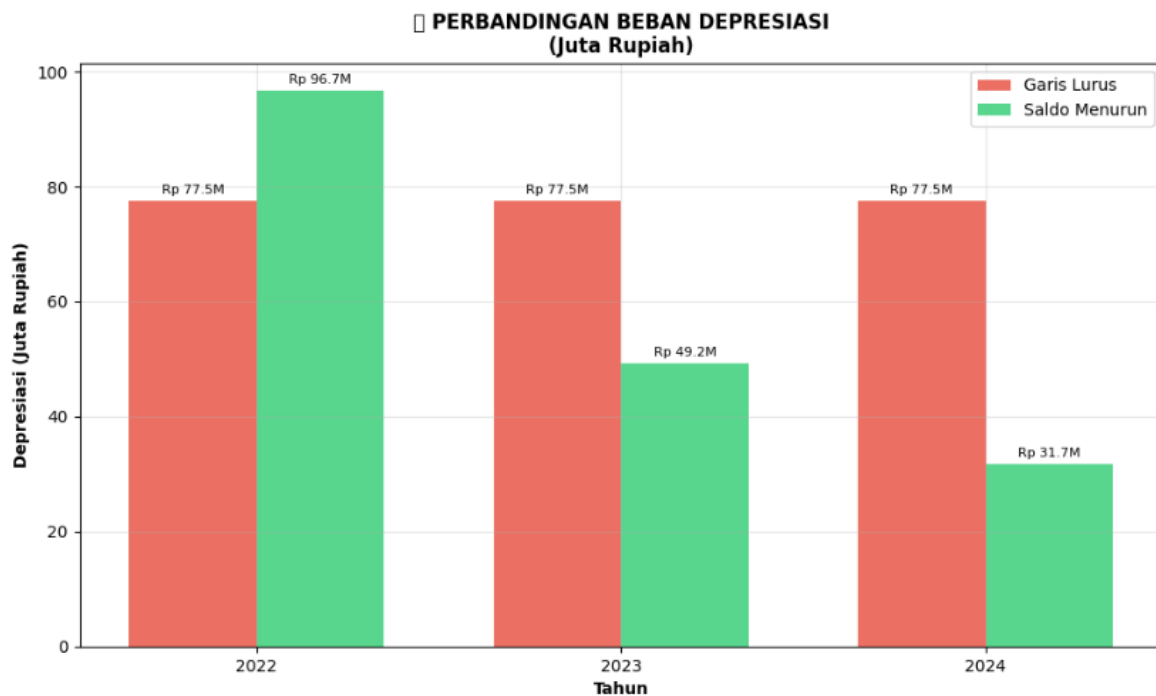
4.1.4 Analisis Profit Margin



Analisis profit margin menunjukkan tren yang kontras antara dua metode perhitungan selama periode 2022-2024. Pada 2022, kedua kategori menunjukkan margin yang relatif rendah dan hampir setara - Garis Lurus 24,8% dan Saldo Menurun 23,3%. Ini mencerminkan kondisi operasional yang masih dalam tahap konsolidasi. Tahun 2023 menjadi periode golden moment dimana kedua margin melonjak tajam. Garis Lurus naik menjadi 40,0% sementara Saldo Menurun mencapai puncak 42,4%. Peningkatan ini menunjukkan efisiensi operasional yang luar biasa dan kemampuan perusahaan dalam mengoptimalkan struktur biaya.

Namun, momentum positif ini tidak bertahan di 2024. Margin Garis Lurus turun menjadi 34,3% meskipun masih di atas level 2022. Saldo Menurun juga mengalami penurunan menjadi 41,7%, namun tetap mempertahankan posisi yang solid. Yang menarik adalah konsistensi Saldo Menurun yang selalu unggul tipis dibanding Garis Lurus, kecuali di tahun 2022. Ini mengindikasikan bahwa metode Saldo Menurun memberikan efisiensi yang lebih baik dalam jangka panjang. Meskipun mengalami fluktuasi, tren margin secara keseluruhan menunjukkan peningkatan dibanding baseline 2022, mencerminkan pembelajaran dan adaptasi yang berkelanjutan.

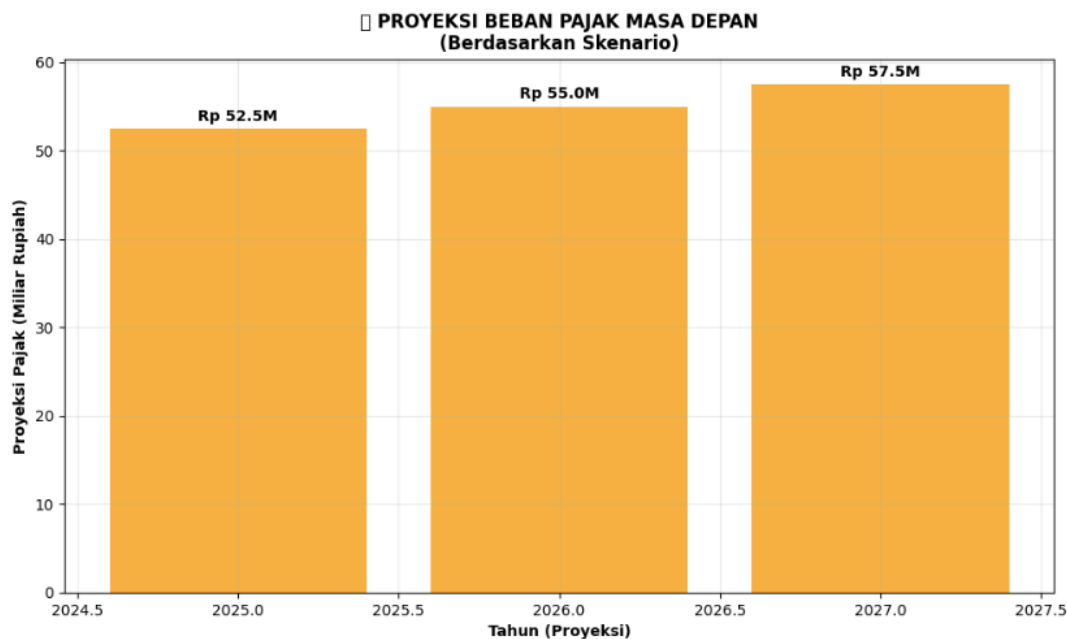
4.1.5 Perbandingan Beban Depresiasi



Pola beban depresiasi menunjukkan karakteristik yang sangat berbeda antara kedua metode selama periode 2022-2024. Metode Garis Lurus menampilkan konsistensi sempurna dengan beban tetap Rp 77,5 miliar setiap tahunnya. Stabilitas ini mencerminkan sifat dasar metode garis lurus yang mendistribusikan nilai aset secara merata sepanjang masa manfaat. Sebaliknya, Saldo Menurun menunjukkan pola penurunan progresif yang dramatis. Dimulai dari puncak Rp 96,7 miliar di 2022, turun signifikan menjadi Rp 49,2 miliar di 2023, dan terus menyusut ke Rp 31,7 miliar di 2024. Penurunan ini sejalan dengan karakteristik accelerated depreciation dimana beban terbesar dibebankan di tahun-tahun awal.

Total beban depresiasi tiga tahun menunjukkan Garis Lurus sebesar Rp 232,5 miliar, sementara Saldo Menurun mencapai Rp 177,6 miliar. Selisih Rp 54,9 miliar ini memberikan keuntungan cash flow yang substansial bagi perusahaan yang menggunakan metode Saldo Menurun. Implikasi strategisnya jelas - Saldo Menurun memberikan tax shield yang lebih besar di tahun-tahun awal, memungkinkan perusahaan memiliki likuiditas lebih baik untuk reinvestasi atau ekspansi. Namun, di tahun-tahun mendatang, beban depresiasi yang rendah akan berdampak pada tax burden yang lebih tinggi.

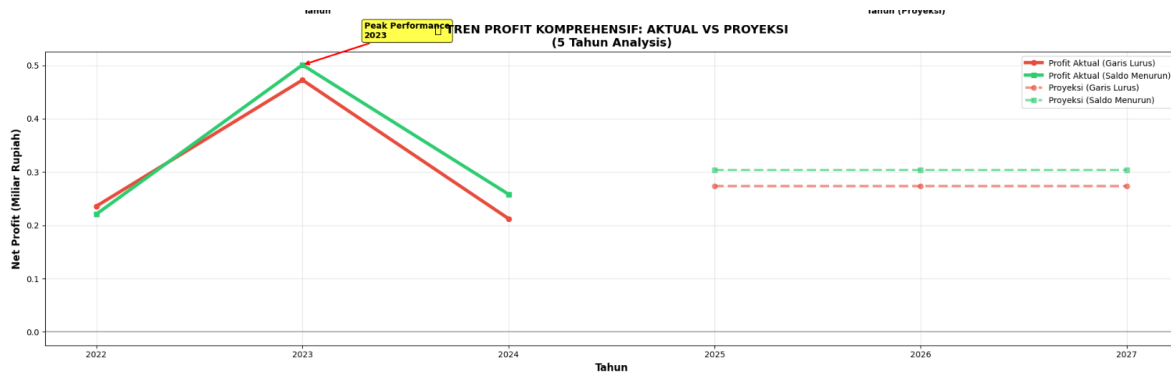
4.1.6 Proyeksi Beban Pajak Masa Depan



Proyeksi beban pajak untuk periode 2025-2027 menunjukkan tren peningkatan yang moderat namun konsisten. Dimulai dari baseline Rp 52,5 miliar di 2025, beban pajak diproyeksikan naik bertahap menjadi Rp 55,0 miliar di 2026 dan mencapai Rp 57,5 miliar di 2027. Pola kenaikan ini mencerminkan proyeksi pertumbuhan bisnis yang realistis dengan rata-rata peningkatan sekitar 4,8% per tahun. Angka ini menunjukkan ekspektasi recovery yang stabil setelah volatilitas tinggi yang dialami dalam periode sebelumnya. Perusahaan tampaknya mengadopsi pendekatan konservatif dalam perencanaan pajak jangka menengah.

Total akumulasi beban pajak tiga tahun diproyeksikan mencapai Rp 165 miliar, yang memberikan gambaran beban finansial yang perlu diantisipasi. Peningkatan gradual ini memberikan ruang bagi perusahaan untuk menyesuaikan strategi cash flow dan perencanaan investasi. Proyeksi yang relatif stabil ini mengindikasikan bahwa perusahaan telah melewati fase turbulensi dan memasuki periode normalisasi. Manajemen dapat menggunakan prediktabilitas ini untuk merancang strategi bisnis jangka menengah yang lebih terukur, termasuk alokasi sumber daya dan rencana ekspansi yang lebih prudent.

4.1.7 Tren Profit Komprehensif dengan Metode Depresiasi



Perbandingan profit komprehensif antara realisasi dan proyeksi menggambarkan perjalanan bisnis yang penuh dinamika. Periode 2022-2024 menunjukkan volatilitas tinggi dengan puncak performa di 2023 yang mencapai sekitar 0,5 miliar rupiah untuk kedua metode perhitungan, kemudian turun drastis ke 0,3 miliar di 2024. Yang menarik adalah konsistensi relatif antara metode Garis Lurus dan Saldo Menurun sepanjang periode aktual. Kedua garis hampir berhimpit, mengindikasikan bahwa perbedaan metodologi tidak memberikan dampak signifikan terhadap profit komprehensif dalam jangka pendek.

Proyeksi 2025-2027 menampilkan kontras yang mencolok dengan stabilitas sempurna pada level 0,31-0,32 miliar rupiah. Garis proyeksi yang mendatar ini mencerminkan pendekatan konservatif manajemen dalam perencanaan jangka menengah. Proyeksi ini tampak realistis mengingat pengalaman volatilitas sebelumnya. Transisi dari era turbulence ke era stability sangat jelas terlihat. Peak performance 2023 yang diikuti koreksi 2024 memberikan pembelajaran berharga bahwa pertumbuhan eksplosif tidak berkelanjutan. Proyeksi flat 2025-2027 menunjukkan fokus pada stabilisasi operasional dan membangun fondasi yang solid untuk pertumbuhan jangka panjang yang lebih sustainable.

BAB V

KESIMPULAN PENGGUNAAN BIGQUERY DAN PYTHON

BigQuery sangat tepat digunakan ketika berhadapan dengan volume data yang besar, seperti jutaan entri transaksi keuangan atau data aset tetap dari berbagai entitas bisnis. Dengan kemampuan pemrosesan cloud yang cepat dan efisien, serta keandalannya dalam menjalankan query SQL yang kompleks, BigQuery menjadi solusi ideal untuk menghitung laba kena pajak, agregasi depresiasi aset, hingga simulasi dampak kebijakan fiskal pada skala nasional atau lintas perusahaan. Keunggulan lainnya adalah data tidak perlu diunduh ke perangkat lokal, sehingga lebih hemat waktu dan sumber daya—terutama bila data sudah berada dalam ekosistem Google Cloud, seperti Google Sheets atau Google Cloud Storage.

Sementara itu, Python sangat bermanfaat pada tahap lanjutan setelah proses pengolahan data selesai dilakukan. Bahasa pemrograman ini memungkinkan analisis mendalam serta visualisasi yang informatif dan menarik. Dengan dukungan berbagai pustaka seperti Pandas untuk pengolahan data, Matplotlib dan Seaborn untuk grafik dua dimensi, serta Plotly untuk visualisasi interaktif dalam bentuk tiga dimensi, Python mampu menyampaikan hasil analisis secara naratif dan visual. Selain itu, Python juga mendukung simulasi skenario, integrasi berbagai sumber data, serta pembuatan laporan otomatis yang bisa disesuaikan dengan kebutuhan manajemen atau pihak terkait.

Oleh karena itu, BigQuery unggul dalam hal pengolahan dan perhitungan data berskala besar secara efisien di cloud, sedangkan Python lebih cocok untuk analisis lanjutan, eksplorasi, dan penyajian data. Dalam praktik terbaik, keduanya saling melengkapi—BigQuery menangani proses awal data besar, lalu Python memaksimalkan hasilnya melalui analisis visual dan interpretatif.