

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama menjalani program magang sebagai Data Analyst Intern di PT Djaya Universal Indonesia, peserta magang secara struktural ditempatkan di bawah Divisi Data Management. Divisi ini berada langsung di bawah koordinasi General Manager Finance, yang kemudian melapor kepada Direktur Utama. Posisi ini secara langsung melibatkan peserta magang dalam kegiatan pengolahan dan analisis data strategis yang berkontribusi terhadap proses pengambilan keputusan di tingkat manajerial, khususnya dalam aspek keuangan dan operasional perusahaan.

Secara lebih rinci, peserta magang menjalankan tugas-tugas harian di bawah arahan dua orang staf Data Management yang berperan sebagai mentor teknis sekaligus penghubung dalam koordinasi pekerjaan. Kedua staf tersebut secara rutin menyampaikan laporan kepada General Manager Finance sebagai atasan langsung dari divisi, sebelum akhirnya diteruskan kepada Direktur untuk evaluasi dan tindak lanjut strategis. Dalam konteks ini, alur koordinasi bersifat dua arah dan terbuka, memungkinkan peserta magang untuk menyampaikan perkembangan pekerjaan, kendala, serta kebutuhan pendukung secara langsung kepada pembimbing melalui sesi diskusi, briefing mingguan, maupun laporan tertulis.

Menjalani peran sebagai Data Analyst Intern di PT Djaya Universal Indonesia selama periode Maret hingga Juni 2025 merupakan sebuah kesempatan untuk mengaplikasikan pengetahuan akademis ke dalam lingkungan industri yang dinamis. Dengan penempatan kerja secara *Work From Office* (WFO) di Pekanbaru, posisi ini ditempatkan secara strategis di dalam Divisi Data Management. Divisi ini merupakan unit kerja yang sangat vital, berfungsi sebagai pusat pengumpulan, pengolahan, dan analisis data untuk mendukung pengambilan keputusan di seluruh perusahaan. Secara hierarkis, Divisi Data Management berada langsung di bawah naungan General Manager Finance, yang menegaskan bahwa setiap hasil analisis

yang dikerjakan bertujuan untuk mendukung proses pengambilan keputusan strategis di tingkat manajerial, terutama dalam aspek keuangan dan investasi.

Alur kerja yang dijalani dirancang secara sistematis untuk mendukung proyek utama, yaitu melakukan analisis dan visualisasi data penjualan, pembelian, dan pengeluaran perusahaan. Proses ini diawali dengan pengumpulan dan eksplorasi data historis perusahaan dari tahun 2022 hingga 2024 yang tersimpan dalam format Microsoft Excel. Data mentah tersebut kemudian diolah melalui tahap pembersihan dan integrasi menggunakan bahasa pemrograman *Python*, di mana data yang tidak lengkap atau duplikat dihilangkan untuk memastikan akurasi analisis. Setelah data bersih, dilakukan analisis mendalam untuk menghitung berbagai metrik kinerja utama sebelum akhirnya diwujudkan dalam bentuk *dashboard* interaktif menggunakan Microsoft Power BI. Dalam menjalankan seluruh rangkaian tugas ini, koordinasi teknis, arahan, dan motivasi harian diberikan oleh seorang Pembimbing Lapangan dari pihak perusahaan. Perusahaan juga menunjukkan dukungannya terhadap kebutuhan akademik dengan memberikan fleksibilitas, asalkan dikoordinasikan dan disetujui terlebih dahulu oleh pembimbing.

3.2 Tugas dan Uraian Kerja Magang

Tugas dan uraian kerja magang difokuskan pada proyek pengembangan *dashboard* analisis data untuk PT Djaya Universal Indonesia. Proses kerja ini berlangsung secara sistematis, dimulai dari pengenalan lingkungan kerja hingga penyerahan proyek akhir.

Table 3.1 Uraian Pelaksanaan Kerja Magang

No.	Jenis Pekerjaan	Uraian Tugas	Tanggal Mulai	Tanggal Selesai
1	Pengenalan Lingkungan dan Software Kerja	Memahami lingkungan kerja, proses bisnis ekspor, dan alur data kantor & gudang.	03/03/2025	08/03/2025
		Mempelajari Excel, <i>Python (Pandas)</i> , Power BI.		
2	Pengumpulan dan Eksplorasi Data	Ekstraksi & restrukturisasi data dari <i>file</i> akuntansi.	10/03/2025	22/03/2025
		Menyusun ulang data menjadi 3 <i>sheet</i> standar.		
3	Pembersihan dan Integrasi Data	Integrasi data 2022–2024 dengan <i>Python</i> .	24/03/2025	12/04/2025
		Tangani missing values & duplikat.		
4	Analisis Data Operasional	Hitung metrik KPI dengan <i>Python</i> .	14/04/2025	22/04/2025
		Analisis keuangan		
5	Visualisasi Data & Dashboard	Buat measure DAX untuk KPI.	23/04/2025	17/05/2025
		Kembangkan grafik dan tabel interaktif.		
6	Evaluasi dan Finalisasi Dashboard	Gabungkan seluruh visual menjadi <i>dashboard</i> tematik	19/05/2025	14/06/2025
7	Serah Terima Kerja	Serahkan <i>file</i> Power BI, skrip <i>Python</i> , data bersih, dan laporan akhir ke pembimbing lapangan.	16/06/2025	21/06/2025
8	Tugas Rutin (Harian)	Update harian data	15/04/2025	28/06/2025

3.2.1 Pengenalan Lingkungan dan Software Kerja

Pada minggu pertama pelaksanaan magang, dilakukan pengenalan terhadap lingkungan kerja serta pemahaman awal terhadap struktur dan sumber data yang digunakan oleh perusahaan. Lingkungan kerja magang adalah area kantor dan gudang PT Djaya Universal Indonesia yang berlokasi di Pekanbaru.

Perusahaan ini bergerak di bidang ekspor hasil pertanian, khususnya produk serat alam seperti sabut kelapa yang diolah dan dikemas sebagai bahan baku industri untuk diekspor ke berbagai negara tujuan.



Gambar 3.1 Gudang PT. Djaya Universal Indonesia

Gudang PT Djaya Universal Indonesia berfungsi sebagai pusat utama penyimpanan dan pengelolaan barang ekspor. Barang ekspor dikemas dalam bentuk bundel besar menggunakan plastik tebal dan diikat dengan kuat untuk memastikan keamanan selama proses pengiriman. Terdapat pula tumpukan karung bahan mentah dan barang jadi yang mencerminkan aktivitas penyortiran, pengeringan, dan pengemasan yang berlangsung setiap hari.

Meskipun tidak secara langsung terlibat dalam pekerjaan di gudang, pemahaman terhadap lingkungan fisik dan proses operasional sangat penting karena data yang dianalisis berasal dari aktivitas lapangan tersebut. Oleh karena itu, beberapa kali dilakukan observasi langsung guna memperoleh gambaran menyeluruh mengenai proses bisnis dan alur ekspor. Secara umum, proses bisnis perusahaan dimulai dari pengumpulan bahan baku sabut kelapa

dari petani atau pemasok lokal. Bahan tersebut kemudian diproses di gudang melalui tahapan penyortiran, pembersihan, pengeringan, dan pengemasan. Setelah siap ekspor, produk dikirim melalui jalur darat atau laut, tergantung pada tujuan pengiriman dan kebutuhan pelanggan.

Sebagai mahasiswa magang yang ditempatkan di Divisi Data Management, perlu untuk mempelajari dan mengelola data dari tahun 2022 - 2024 yang mendukung aktivitas ekspor, di antaranya:

1. Data penjualan: berisi informasi produk, jumlah, harga satuan dan total, serta negara tujuan ekspor.
2. Data pembelian: mencatat transaksi pembelian bahan baku dari supplier lokal.
3. Data pengeluaran: memuat biaya operasional seperti transportasi, gaji, dan pembelian peralatan kerja.

Dengan memahami secara langsung proses bisnis perusahaan, pengelolaan dan analisis data dapat dilakukan secara kontekstual dan relevan. Hal ini turut mendukung penyusunan laporan serta visualisasi data yang akurat untuk membantu proses pengambilan keputusan perusahaan.

Dalam pelaksanaan kerja magang di Divisi Data Management, pendekatan analitis yang terstruktur diterapkan dengan mengintegrasikan beberapa perangkat lunak. Untuk menjawab kebutuhan analisis data yang komprehensif, mulai dari pembersihan data mentah hingga penyajian laporan akhir, alur kerja bergantung pada tiga pilar teknologi utama: Microsoft Excel untuk penanganan data awal, *Python* untuk pemrosesan dan analisis mendalam, serta Microsoft Power BI untuk visualisasi bisnis. Kombinasi ini memungkinkan pengolahan data yang efisien, dapat direproduksi, dan menghasilkan wawasan yang berdampak.

Tahap pertama dalam alur kerja analisis adalah eksplorasi dan persiapan data menggunakan Microsoft Excel. Perangkat lunak ini dimanfaatkan untuk

membuka *file* data mentah yang disediakan oleh perusahaan, yang umumnya dalam format *spreadsheet*. Di tahap ini, Excel sangat berguna untuk melakukan inspeksi visual, memahami struktur kolom, serta melakukan pembersihan data tingkat dasar seperti mengatasi format yang tidak konsisten atau kesalahan input sederhana [5]. Namun, untuk menangani volume data yang mencakup periode beberapa tahun dan melakukan transformasi yang kompleks, diperlukan kapabilitas yang lebih tinggi [6].

Oleh karena itu, tahap pemrosesan inti dilakukan dengan menggunakan bahasa pemrograman *Python*, yang dikenal dengan ekosistem analisis datanya yang kuat [7]. Secara spesifik, *library Pandas* menjadi alat utama yang digunakan untuk tugas-tugas manipulasi data secara terprogram. Dengan *Pandas*, data dapat dibersihkan secara efisien, tabel dari sumber yang berbeda dapat digabungkan, serta agregasi dilakukan untuk menghitung metrik performa. Penggunaan *Python* memastikan bahwa proses analisis dapat diulang dan diaudit dengan mudah, yang merupakan keunggulan signifikan dibandingkan proses manual [8].

Tahap akhir dari alur kerja ini adalah mengubah data yang telah diolah menjadi wawasan visual melalui Microsoft Power BI. Platform intelijen bisnis ini digunakan untuk merancang *dashboard* interaktif yang menyajikan temuan-temuan kunci dari analisis. Visualisasi seperti grafik tren penjualan dan diagram rincian biaya. *Dashboard* ini tidak hanya berfungsi sebagai laporan statis, tetapi juga sebagai alat eksplorasi bagi manajemen, yang memungkinkan mereka untuk memfilter dan menelusuri data secara mandiri untuk mendukung pengambilan keputusan strategis yang berbasis bukti [9], [10].

3.2.2 Pengumpulan dan Eksplorasi Data

Langkah awal yang menjadi fondasi dalam keseluruhan proses analisis data adalah tahap pengumpulan, eksplorasi, dan restrukturisasi data mentah. Berbeda dari format data yang telah disederhanakan, data historis asli yang disediakan perusahaan untuk tahun 2022, 2023, dan 2024 merupakan *file*

kerja akuntansi yang sangat kompleks. Setiap *file* Excel tahunan tersebut berisi berbagai macam *sheet* yang mencatat seluruh aktivitas keuangan secara terperinci, seperti Jurnal Kas, Jurnal Pengeluaran, Jurnal Piutang, Persediaan, Invoice, dan Laporan keuangan.

Tgl	Metode Pembayaran	Jenis Pengeluaran	Deskripsi	Jumlah
02-Sep-22	Mestika	Pembayaran Operational		
05-Sep-22	Mestika	Pembayaran Operational		
17-Sep-22	Mestika	Pembayaran Operational		
18-Sep-22	Mestika	Pembayaran Operational		
18-Sep-22	Mestika	Pembayaran Operational		
18-Sep-22	Mestika	Pembayaran Operational		
15-Oct-22	Mestika	Pembayaran Operational		
09-Oct-22	Mestika	Pembayaran Operational		
15-Oct-22	Mestika	Pembayaran Operational		
15-Oct-22	Mestika	Pembayaran Operational		
25-Oct-22	Mestika	Pembayaran Operational		
25-Oct-22	Mestika	Pembayaran Operational		
07-Nov-22	Mestika	Pembayaran Operational		
07-Nov-22	Mestika	Pembayaran Operational		
13-Nov-22	Mestika	Pembayaran Operational		
16-Nov-22	Mestika	Pembayaran Operational		
17-Nov-22	Mestika	Pembayaran Operational		
18-Nov-22	Mestika	Pembayaran Operational		

Gambar 3.2 Contoh Data Mentah

Berikut adalah data mentah yang berupa data konfidensial dari PT. Djaya Universal Indonesia. Melihat kompleksitas dan fragmentasi data sumber tersebut, tugas pertama yang krusial adalah melakukan ekstraksi dan restrukturisasi data secara manual. Proses ini menuntut pemahaman terhadap alur akuntansi perusahaan untuk dapat menavigasi berbagai *sheet* jurnal dan laporan, kemudian mengidentifikasi serta memilah data yang relevan secara spesifik untuk tiga pilar utama analisis: pengeluaran, penjualan, dan pembelian. Sebagai contoh, informasi untuk pengeluaran perlu ditarik dari beberapa *sheet* seperti Jurnal Pengeluaran dan Jurnal Kas, sementara data penjualan perlu dikonsolidasikan dari Penjualan, Invoice, dan Jurnal Piutang.

Tgl	Metode Pembayaran	Jenis Pengeluaran	Deskripsi	Jumlah
02-Sep-22	Mestika			
05-Sep-22	Mestika			
17-Sep-22	Mestika			
18-Sep-22	Mestika			
18-Sep-22	Mestika			
18-Sep-22	Mestika			
15-Oct-22	Mestika			
09-Oct-22	Mestika			
15-Oct-22	Mestika			
15-Oct-22	Mestika			
25-Oct-22	Mestika			
25-Oct-22	Mestika			
07-Nov-22	Mestika			
07-Nov-22	Mestika			
13-Nov-22	Mestika			
16-Nov-22	Mestika			
17-Nov-22	Mestika			
18-Nov-22	Mestika			
21-Nov-22	Mestika			
24-Nov-22	Mestika			
04-Nov-22	Mestika			
02-Nov-22	Mestika			
10-Nov-22	Mestika			
07-Dec-22	Mestika			
09-Dec-22	Mestika			
11-Dec-22	Mestika			
12-Dec-22	Mestika			

Gambar 3.3 Contoh Processed Data

Berikut adalah data konfidensial yang telah di proses dari PT. Djaya Universal Indonesia. Hasil dari proses ekstraksi manual ini adalah pembuatan sebuah struktur *file* Excel baru yang telah disederhanakan. Proses restrukturisasi ini diterapkan secara individual untuk setiap tahunnya. Sebagai contoh konkret, *file* kerja Data 2022.xlsx yang rumit diubah menjadi sebuah *file* baru yang lebih sederhana khusus untuk tahun 2022. *File* baru ini hanya berisi tiga *sheet* yang telah distandarisasi, yaitu Pengeluaran, Penjualan, dan Pembelian. Prosedur yang sama diulangi untuk data tahun 2023 dan 2024, sehingga menghasilkan satu set *file* tahunan yang bersih dan konsisten. Persiapan ini menjadi langkah fundamental yang wajib dilakukan sebelum beralih ke tahap analisis menggunakan perangkat lunak, karena dengan data yang telah terstruktur ulang, proses penggabungan dan analisis lanjutan menggunakan *Python* dapat berjalan secara lebih efisien dan terfokus.

3.2.3 Pembersihan dan Integrasi Data

Langkah selanjutnya adalah mempersiapkan serta menggabungkan seluruh data dari berbagai *file* Excel yang mencakup tiga tahun terakhir, yaitu tahun 2022, 2023, dan 2024. Setiap *file* Excel yang terdiri dari tiga *sheet* utama, yaitu Pengeluaran, Penjualan, dan Pembelian. Ketiga *sheet* tersebut perlu diproses dan dikompilasi terlebih dahulu ke dalam satu struktur data yang terpisah untuk masing-masing jenis transaksi, guna memudahkan proses analisis lebih lanjut.

Proses ini dilakukan dengan menggunakan bahasa pemrograman *Python* dan *library Pandas* yang dikenal handal dalam melakukan manipulasi dan analisis data dalam format tabel. Pada tahap awal pengolahan data, dilakukan proses ekstraksi dari setiap *file* Excel yang berisi tiga *sheet* utama. Untuk mempercepat dan menyederhanakan proses pembacaan data secara otomatis, dibuat sebuah fungsi *Python* bernama `read_selected_sheets`. Fungsi ini dikembangkan untuk membaca semua *sheet* yang telah ditentukan dari satu *file* Excel hanya dengan satu kali pemanggilan fungsi. Dengan pendekatan ini, efisiensi kerja meningkat karena tidak perlu menulis kode berulang untuk setiap *sheet* dan *file*.

```
[2]: sheet_names = ['Pengeluaran', 'Penjualan', 'Pembelian']

def read_selected_sheets(file_path, file_label):
    data = {}
    for sheet in sheet_names:
        try:
            df = pd.read_excel(file_path, sheet_name=sheet)
            df['Tahun'] = file_label
            df['Sheet'] = sheet
            data[sheet] = df
        except Exception as e:
            print(f"Gagal membaca sheet '{sheet}' dari {file_path}: {e}")
    return data

data_2022 = read_selected_sheets('2022.xlsx', '2022')
data_2023 = read_selected_sheets('2023.xlsx', '2023')
data_2024 = read_selected_sheets('2024.xlsx', '2024')

data_2022
```

Gambar 3.4 Kode Python untuk Membaca Data dari Beberapa Sheet Excel

Gambar 3.4 di atas menunjukkan sebuah fungsi *Python* bernama `read_selected_sheets` yang dirancang untuk membaca beberapa *sheet* spesifik dari *file* Excel. Untuk setiap *sheet* yang berhasil dibaca, fungsi ini secara otomatis menambahkan dua kolom baru: kolom 'Tahun' untuk mengidentifikasi tahun asal data dan kolom 'Sheet' untuk mengidentifikasi nama *sheet* asalnya. Fungsi ini kemudian dipanggil secara berulang untuk memproses *file* Excel dari tahun 2022, 2023, dan 2024, sehingga menyederhanakan proses impor dan pengorganisasian data dari beberapa *file* ke dalam struktur data yang siap untuk dianalisis.

```
[2]: {'Pengeluaran':
      0      NaT      NaN      NaN
      1 2022-09-02  Mestika      NaN
      2 2022-09-05  Mestika      NaN
      3 2022-09-17  Mestika      NaN
      4 2022-09-18  Mestika      NaN
      ..
      60 2022-12-23  Mestika      NaN
      61 2022-12-23  Mestika      NaN
      62 2022-12-23  Mestika      NaN
      63 2022-12-23  Mestika      NaN
      64 2022-12-23  Mestika      NaN

      Deskripsi      Jumlah Tahun      Sheet
      0      NaN      NaN 2022  Pengeluaran
      1
      2
      3
      4
      ..
      60
      61
      62
      63
      64

[65 rows x 7 columns],
'Penjualan':
      0      NaT      NaN      NaN      NaN      NaN      NaN      NaN
      1 2022-09-21  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      2 2022-10-31  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      3 2022-11-16  Persediaan  LS  Lidi Sawit  Penjualan  Lidi sawit
      4 2022-11-23  Persediaan  LS  Lidi Sawit  Penjualan  Lidi sawit
      5 2022-11-23  Persediaan  LS  Lidi Sawit  Penjualan  Lidi sawit
      6 2022-12-07  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      7 2022-12-07  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      8 2022-12-08  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      9 2022-12-12  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      10 2022-12-12  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      11 2022-12-22  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      12 2022-12-22  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      13 2022-12-27  Persediaan  LN  Lidi Nipah  Penjualan  Lidi nipah
      14 2022-12-29  Persediaan  LS  Lidi Sawit  Penjualan  Lidi sawit
```

Gambar 3.5 Data Hasil Pembacaan Sheet

Gambar 3.5 diatas terdapat data konfidensial yang menunjukkan hasil dari pemanggilan fungsi tersebut disimpan dalam tiga variabel terpisah, yaitu `data_2022`, `data_2023`, dan `data_2024`. Masing-masing variabel berisi tiga *DataFrame* yang mewakili *sheet* Pengeluaran, Penjualan, dan Pembelian. Sebagai contoh, pada *sheet* Pengeluaran, terdapat data transaksi yang meliputi kolom “Tgl”, “Metode Pembayaran”, “Jenis Pengeluaran”, “Deskripsi”, dan

“Jumlah”. Kolom-kolom ini mencerminkan berbagai aktivitas pengeluaran perusahaan, mulai dari biaya transportasi dan logistik, biaya perawatan kendaraan, hingga pembelian barang dan material penunjang kegiatan operasional.

Lalu, pada *sheet* Penjualan, data yang tersedia mencakup informasi produk yang dijual, dengan atribut pendukung seperti “Tgl”, “Jenis”, “Kode”, “Produk”, “Deskripsi”, “Kuantitas”, “Satuan”, dan lainnya. Data penjualan ini memiliki peran penting dalam memahami performa penjualan perusahaan dari tahun ke tahun, serta dalam mengevaluasi kontribusi setiap jenis produk terhadap total pendapatan.

Sementara itu, pada *sheet* Pembelian, data yang tersedia mencakup informasi produk yang dibeli, dengan atribut pendukung seperti “Tgl”, “Jenis”, “Kode”, “Produk”, “Deskripsi”, “Kuantitas”, “Satuan”, “Harga”, dan lainnya. Data pembelian ini memiliki peran penting untuk mengetahui ketersediaan atau pergerakan stok produk dari tahun ke tahun. Melalui data ini, perusahaan dapat mengevaluasi pola pembelian, mengidentifikasi kebutuhan pengadaan barang berdasarkan frekuensi dan volume pembelian, serta menganalisis perubahan harga dari masing-masing produk. Dengan menggabungkan data pembelian dari tahun 2022 hingga 2024 ke dalam satu struktur data terpusat, dapat dilakukan analisis historis secara menyeluruh, seperti melihat tren pembelian musiman, membandingkan harga antar tahun, hingga menilai kontribusi pembelian terhadap total pengeluaran perusahaan.

```
[5]: # Gabungkan semua data berdasarkan sheet
df_pengeluaran = pd.concat([data_2022['Pengeluaran'], data_2023['Pengeluaran'], data_2024['Pengeluaran']], ignore_index=True)
df_penjualan = pd.concat([data_2022['Penjualan'], data_2023['Penjualan'], data_2024['Penjualan']], ignore_index=True)
df_pembelian = pd.concat([data_2022['Pembelian'], data_2023['Pembelian'], data_2024['Pembelian']], ignore_index=True)

df_pengeluaran
```

Gambar 3.6 Kode Python untuk Menggabung Data Berdasarkan Nama Sheet

Gambar 3.6 di atas menunjukkan kode yang digunakan untuk menggabungkan data dari beberapa tahun (2022, 2023, dan 2024) yang sebelumnya terpisah ke dalam tiga *DataFrame* utama. Dengan menggunakan fungsi `pd.concat`, data Pengeluaran dari setiap tahun disatukan secara vertikal

menjadi satu *DataFrame* `df_pengeluaran`. Proses yang sama diulangi untuk data Penjualan dan Pembelian. Penggunaan argumen `ignore_index=True` bertujuan untuk membuat indeks baru yang berurutan pada *DataFrame* gabungan, sehingga memastikan setiap baris memiliki indeks yang unik. Langkah ini penting untuk mengonsolidasikan semua data historis agar dapat dianalisis secara keseluruhan.

[5]:

	Tgl	Metode Pembayaran	Jenis Pengeluaran	Deskripsi	Jumlah	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Tahun	Sheet
0	NaT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2022	Pengeluaran
1	2022-09-02	Mestika	Transportasi dan Logistik			NaN	NaN	NaN	NaN	2022	Pengeluaran
2	2022-09-05	Mestika	Transportasi dan Logistik			NaN	NaN	NaN	NaN	2022	Pengeluaran
3	2022-09-17	Mestika	Transportasi dan Logistik			NaN	NaN	NaN	NaN	2022	Pengeluaran
4	2022-09-18	Mestika	Transportasi dan Logistik			NaN	NaN	NaN	NaN	2022	Pengeluaran
...
816	2024-12-30	Uang Tunai	Operasional Gudang			NaN	NaN	NaN	NaN	2024	Pengeluaran
817	2024-12-30	Uang Tunai	Operasional Gudang			NaN	NaN	NaN	NaN	2024	Pengeluaran
818	2024-12-30	Uang Tunai	Operasional Gudang			NaN	NaN	NaN	NaN	2024	Pengeluaran
819	2024-12-30	Uang Tunai	Operasional Harian			NaN	NaN	NaN	NaN	2024	Pengeluaran
820	2024-12-30	Uang Tunai	Barang dan Material			NaN	NaN	NaN	NaN	2024	Pengeluaran

821 rows x 11 columns

Gambar 3.7 Contoh *DataFrame* 'df_pengeluaran' Setelah Penggabungan Data

Gambar 3.7 di atas menunjukkan data konfidensial yang merupakan hasil penggabungan seluruh informasi pengeluaran perusahaan dari tahun 2022 hingga 2024. Data ini mencakup berbagai jenis pengeluaran yang terjadi selama periode tersebut dan disusun berdasarkan *sheet* yang bersumber dari masing-masing tahun. Namun, beberapa kolom seperti Unnamed: 5, Unnamed: 6, Unnamed: 7, dan Unnamed: 8 tidak memiliki header yang bermakna dan seluruh nilainya adalah NaN (kosong).

```
[6]: df_penjualan
```

	Tgl	Jenis	Kode	Produk	Deskripsi	Kuantitas	Satuan	HPP satuan	HPP	Harga satuan	Total Harga	Harga yang dibayarkan	Pembayaran	Tahun
0	NaT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Uang Tunai	2022
1	2022-09-21	Persediaan	LN	Lidi Nipah	Penjualan Lidi nipah									2022
2	2022-10-31	Persediaan	LN	Lidi Nipah	Penjualan Lidi nipah									2022
3	2022-11-16	Persediaan	LS	Lidi Sawit	Penjualan Lidi sawit									2022
4	2022-11-23	Persediaan	LS	Lidi Sawit	Penjualan Lidi sawit									2022
...
90	2024-12-09	Persediaan	LB	Lidi bonggol	Penjualan lidi bonggol									2024
91	2024-12-12	Persediaan	LB	Lidi bonggol	Penjualan lidi bonggol									2024
92	2024-12-12	Persediaan	LP	Lidi potong	Penjualan lidi potong									2024
93	2024-12-23	Persediaan	LB	Lidi bonggol	Penjualan lidi bonggol									2024
94	2024-12-23	Persediaan	LB	Lidi bonggol	Penjualan lidi bonggol									2024

95 rows x 15 columns

Gambar 3.8 Contoh DataFrame 'df_penjualan' Setelah Penggabungan Data

Gambar 3.8 di atas menunjukkan data konfidensial yang berisi informasi penjualan produk perusahaan selama tiga tahun terakhir yang juga berasal dari penggabungan *sheet* masing-masing tahun. Dengan menggabungkan data ini, perusahaan dapat melakukan analisis tren penjualan dan kinerja produk dalam periode tertentu. Data ini menjadi dasar untuk menilai perkembangan dan efektivitas penjualan dari waktu ke waktu.



```
[7]: df_pembelian
```

	Tgl	Jenis	Kode	Produk	Deskripsi	Kuantitas	Satuan	Harga	Total Harga	Harga yang dibayarkan	Pembayaran	Tahun	Sheet
0	NaT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Uang Tunai	2022	Pembelian
1	2022-07-17	Persediaan	LN	Lidi Nipah								2022	Pembelian
2	2022-07-18	Persediaan	LN	Lidi Nipah								2022	Pembelian
3	2022-07-18	Persediaan	LN	Lidi Nipah								2022	Pembelian
4	2022-07-20	Persediaan	LN	Lidi Nipah								2022	Pembelian
...
860	2024-12-29	Persediaan	LP	Lidi potong								2024	Pembelian
861	2024-12-31	Persediaan	LP	Lidi potong								2024	Pembelian
862	2024-12-31	Persediaan	LP	Lidi potong								2024	Pembelian
863	2024-12-31	Persediaan	LP	Lidi potong								2024	Pembelian
864	2024-12-31	Persediaan	LK	Lidi kelapa								2024	Pembelian

865 rows x 13 columns

Gambar 3.9 Contoh DataFrame 'df_pembelian' Setelah Penggabungan Data

Gambar 3.9 di atas data konfidensial yang menunjukkan data pembelian mencakup seluruh transaksi pembelian barang atau bahan baku dari tahun 2022 hingga 2024 yang diperoleh dari penggabungan *sheet* tahunan. Tujuan penggabungan ini adalah untuk mendapatkan gambaran menyeluruh tentang aktivitas pembelian perusahaan. Selanjutnya, data ini dapat digunakan untuk menganalisis manajemen persediaan dan efisiensi pembelian perusahaan.

```
[8]: df_pengeluaran = df_pengeluaran.loc[:, ~df_pengeluaran.columns.str.contains("^Unnamed")]
df_pengeluaran
```

	Tgl	Metode Pembayaran	Jenis Pengeluaran	Deskripsi	Jumlah	Tahun	Sheet
0	NaT	NaN	NaN	NaN	NaN	2022	Pengeluaran
1	2022-09-02		Mestika			2022	Pengeluaran
2	2022-09-05		Mestika			2022	Pengeluaran
3	2022-09-17		Mestika			2022	Pengeluaran
4	2022-09-18		Mestika			2022	Pengeluaran
...
816	2024-12-30	Uang Tunai				2024	Pengeluaran
817	2024-12-30	Uang Tunai				2024	Pengeluaran
818	2024-12-30	Uang Tunai				2024	Pengeluaran
819	2024-12-30	Uang Tunai				2024	Pengeluaran
820	2024-12-30	Uang Tunai				2024	Pengeluaran

821 rows x 7 columns

Gambar 3.10 DataFrame df_pengeluaran Setelah Menghapus Kolom Unnamed

Gambar 3.10 di atas data konfidensial yang menunjukkan hasil proses pembersihan data pada *DataFrame* df_pengeluaran, di mana kolom-kolom

yang tidak memiliki nama atau dinamai “Unnamed” telah dihapus. Kolom-kolom tersebut umumnya berisi nilai kosong (NaN) dan tidak memuat informasi penting, sehingga perlu dihilangkan untuk menyederhanakan struktur data. Proses ini dilakukan menggunakan fungsi `.loc` dan metode `.str.contains()` untuk menyaring hanya kolom yang relevan. Setelah pembersihan, *DataFrame* menyisakan tujuh kolom utama yang diperlukan dalam analisis, yaitu: Tgl, Metode Pembayaran, Jenis Pengeluaran, Deskripsi, Jumlah, Tahun, dan *Sheet*. Dengan data yang lebih rapi dan terstruktur ini, analisis lanjutan dapat dilakukan dengan lebih mudah dan hasilnya menjadi lebih akurat.

```
[12]: print("\n=== Pengeluaran ===")
      print(df_pengeluaran.columns)

      print("\n=== Penjualan ===")
      print(df_penjualan.columns)

      print("\n=== Pembelian ===")
      print(df_pembelian.columns)

      === Pengeluaran ===
      Index(['Tgl', 'Metode Pembayaran', 'Jenis Pengeluaran', 'Deskripsi', 'Jumlah',
            'Tahun', 'Sheet'],
            dtype='object')

      === Penjualan ===
      Index(['Tgl', 'Jenis', 'Kode', 'Produk', 'Deskripsi', 'Kuantitas', 'Satuan',
            'HPP satuan', 'HPP', 'Harga satuan', 'Total Harga',
            'Harga yang dibayarkan', 'Pembayaran', 'Tahun', 'Sheet'],
            dtype='object')

      === Pembelian ===
      Index(['Tgl', 'Jenis', 'Kode', 'Produk', 'Deskripsi', 'Kuantitas', 'Satuan',
            'Harga', 'Total Harga', 'Harga yang dibayarkan', 'Pembayaran', 'Tahun',
            'Sheet'],
            dtype='object')
```

Gambar 3.11 Daftar Kolom

Gambar 3.11 di atas menunjukkan kode yang berfungsi untuk memahami struktur data yang digunakan dalam proses analisis, dilakukan eksplorasi awal terhadap nama-nama kolom pada masing-masing *DataFrame*, yaitu `df_pengeluaran`, `df_penjualan`, dan `df_pembelian`. *DataFrame* `df_pengeluaran` berisi informasi terkait pengeluaran perusahaan, dengan kolom-kolom seperti:

- a) Tgl: Tanggal terjadinya transaksi pengeluaran.

- b) Metode Pembayaran: Cara pembayaran yang digunakan, seperti uang tunai atau transfer bank.
- c) Jenis Pengeluaran: Kategori dari pengeluaran, misalnya transportasi, operasional, atau pembelian bahan.
- d) Deskripsi: Keterangan lebih lanjut mengenai pengeluaran yang dilakukan.
- e) Jumlah: Besarnya nominal dana yang dikeluarkan dalam transaksi.
- f) Tahun: Tahun terjadinya pengeluaran.
- g) *Sheet*: Nama *sheet* sumber data, digunakan untuk menandai bahwa data berasal dari *sheet* "Pengeluaran".

Lalu, `df_penjualan` memuat data hasil penjualan produk perusahaan, dengan atribut lebih kompleks seperti:

- a) Tgl: Tanggal transaksi penjualan dilakukan.
- b) Jenis: Jenis barang yang dijual berdasarkan kategori stok.
- c) Kode: Kode unik produk untuk identifikasi.
- d) Produk: Nama produk yang dijual.
- e) Deskripsi: Rincian tambahan mengenai produk atau transaksi penjualan.
- f) Kuantitas: Jumlah produk yang dijual.
- g) Satuan: Satuan pengukuran produk (misalnya: Kg).
- h) HPP satuan: Harga pokok produksi per satuan.
- i) HPP: Total biaya produksi untuk seluruh kuantitas produk yang dijual.
- j) Harga satuan: Harga jual per satuan produk.
- k) Total Harga: Total nilai penjualan (kuantitas x harga satuan).
- l) Harga yang dibayarkan: Nominal aktual yang diterima dari pembeli.
- m) Pembayaran: Metode pembayaran yang dilakukan pembeli.
- n) Tahun: Tahun terjadinya penjualan.
- o) *Sheet*: Nama *sheet* sumber data, menandakan bahwa data berasal dari *sheet* "Penjualan".

Terakhir, `df_pembelian` mencakup data transaksi pembelian bahan atau produk, dengan struktur kolom yang serupa dengan `df_penjualan`, meliputi:

- a) Tgl: Tanggal transaksi pembelian dilakukan.
- b) Jenis: Jenis barang yang dibeli berdasarkan klasifikasi stok.
- c) Kode: Kode unik produk yang dibeli.
- d) Produk: Nama produk atau bahan yang dibeli.
- e) Deskripsi: Rincian tambahan mengenai produk atau keperluan pembelian.
- f) Kuantitas: Jumlah barang yang dibeli.
- g) Satuan: Satuan pengukuran barang (misalnya: Kg).
- h) Harga: Harga satuan dari barang yang dibeli.
- i) Total Harga: Total nilai pembelian (kuantitas x harga satuan).
- j) Harga yang dibayarkan: Nominal aktual yang dibayarkan ke pihak penjual.
- k) Pembayaran: Metode pembayaran untuk pembelian.
- l) Tahun: Tahun terjadinya pembelian.
- m) *Sheet*: Nama *sheet* sumber data, menandakan bahwa data berasal dari *sheet* "Pembelian".

```
[9]: print("\n=== Pengeluaran ===")
      print(df_pengeluaran.dtypes)

      print("\n=== Penjualan ===")
      print(df_penjualan.dtypes)

      print("\n=== Pembelian ===")
      print(df_pembelian.dtypes)
```

Gambar 3.12 Kode Python untuk Menampilkan Tipe Data

Gambar 3.12 di atas menunjukkan kode program yang ditampilkan bertujuan untuk menampilkan tipe data dari setiap kolom pada tiga *DataFrame*, yaitu `df_pengeluaran`, `df_penjualan`, dan `df_pembelian`. Dengan menggunakan fungsi `.dtypes` dari pustaka *Pandas*, setiap kolom pada ketiga *DataFrame* tersebut diperiksa tipe datanya, apakah berupa *object*, angka *float64*,

datetime64, atau jenis lainnya. Langkah ini penting dilakukan dalam tahap eksplorasi data karena bertujuan untuk memastikan bahwa setiap kolom memiliki tipe data yang sesuai dengan konteks analisis yang akan dilakukan.

Sebagai contoh, kolom “Tgl” seharusnya bertipe *datetime64*, sedangkan kolom nilai seperti “Jumlah”, “Harga”, atau “Total Harga” idealnya bertipe *float64*. Jika ditemukan ketidaksesuaian tipe data, maka perlu dilakukan proses pembersihan dan konversi tipe data sebelum masuk ke tahap analisis lebih lanjut. Validasi tipe data ini menjadi salah satu bagian penting dalam praktik *data preprocessing* selama pelaksanaan kerja magang.

```

=== Pengeluaran ===
Tgl          datetime64[ns]
Metode Pembayaran      object
Jenis Pengeluaran      object
Deskripsi             object
Jumlah               float64
Tahun                object
Sheet                object
dtype: object

=== Penjualan ===
Tgl          datetime64[ns]
Jenis          object
Kode          object
Produk        object
Deskripsi     object
Kuantitas     float64
Satuan        object
HPP satuan   float64
HPP           float64
Harga satuan float64
Total Harga   float64
Harga yang dibayarkan float64
Pembayaran    object
Tahun         object
Sheet         object
dtype: object

=== Pembelian ===
Tgl          datetime64[ns]
Jenis          object
Kode          object
Produk        object
Deskripsi     object
Kuantitas     float64
Satuan        object
Harga         float64
Total Harga   float64
Harga yang dibayarkan float64
Pembayaran    object
Tahun         object
Sheet         object
dtype: object

```

Gambar 3.13 Tipe Data setiap Kolom

Gambar 3.13 di atas menunjukkan hasil output menunjukkan bahwa kolom tanggal (Tgl) pada ketiga *DataFrame* memiliki tipe data *datetime64*, yang berarti telah dikenali sebagai format tanggal. Sebagian besar kolom lainnya bertipe *object*, yang umumnya merepresentasikan data teks, seperti kolom nama produk, deskripsi, atau metode pembayaran. Sementara itu, kolom yang

mengandung nilai numerik seperti jumlah, kuantitas, harga, dan total harga memiliki tipe data *float64*, yang menandakan bahwa data tersebut berupa angka desimal. Pemeriksaan tipe data ini penting dilakukan untuk memastikan setiap kolom berada dalam format yang sesuai sebelum dilakukan analisis data lebih lanjut.

```
[11]: print(df_pengeluaran.isnull().sum())
```

Tgl	3
Metode Pembayaran	3
Jenis Pengeluaran	3
Deskripsi	3
Jumlah	3
Tahun	0
Sheet	0

dtype: int64

Gambar 3.14 Cek Nilai Kosong pada setiap Kolom DataFrame *df_pengeluaran*

Gambar 3.14 di atas menunjukkan hasil pemeriksaan terhadap nilai kosong pada *DataFrame* *df_pengeluaran* dengan menggunakan fungsi *isnull().sum()*. Dari hasil tersebut diketahui bahwa terdapat 3 nilai kosong pada masing-masing kolom *Tgl*, *Metode Pembayaran*, *Jenis Pengeluaran*, *Deskripsi*, dan *Jumlah*. Sementara itu, kolom *Tahun* dan *Sheet* tidak memiliki nilai kosong.

```
[12]: df_pengeluaran = df_pengeluaran.dropna()
print(df_pengeluaran.isnull().sum())
```

Tgl	0
Metode Pembayaran	0
Jenis Pengeluaran	0
Deskripsi	0
Jumlah	0
Tahun	0
Sheet	0

dtype: int64

Gambar 3.15 Hasil Pengecekan Nilai Kosong Setelah Penghapusan Baris

Gambar 3.15 di atas menunjukkan proses pembersihan data pada *DataFrame* *df_pengeluaran* dengan menggunakan fungsi *dropna()*. Fungsi ini digunakan untuk menghapus seluruh baris yang mengandung nilai kosong. Setelah fungsi ini dijalankan, dilakukan pengecekan ulang terhadap jumlah nilai kosong di setiap kolom menggunakan *isnull().sum()*.

Hasilnya menunjukkan bahwa seluruh kolom sudah tidak memiliki nilai kosong. Langkah ini penting dilakukan agar analisis data dapat berjalan dengan akurat dan tidak terganggu oleh data yang tidak lengkap.

```
[13]: print(df_penjualan.isnull().sum())
```

Tgl	3
Jenis	3
Kode	3
Produk	3
Deskripsi	3
Kuantitas	3
Satuan	3
HPP satuan	3
HPP	3
Harga satuan	3
Total Harga	3
Harga yang dibayarkan	3
Pembayaran	0
Tahun	0
Sheet	0
dtype: int64	

Gambar 3.16 Mengecek Jumlah Nilai Kosong

Gambar 3.16 diatas menunjukkan hasil pemeriksaan data kosong pada *DataFrame* `df_penjualan`. Berdasarkan *output* dari perintah `df_penjualan.isnull().sum()`, terdapat nilai kosong sebanyak 3 baris pada hampir seluruh kolom, seperti Tgl, Jenis, Kode, Produk, Deskripsi, Kuantitas, Satuan, HPP satuan, HPP, Harga satuan, Total Harga, dan Harga yang dibayarkan. Sementara itu, kolom Pembayaran, Tahun, dan *Sheet* tidak memiliki nilai kosong.

Hal ini menunjukkan bahwa sebagian data pada awal atau bagian tertentu dari dataset belum terisi secara lengkap, sehingga perlu dilakukan pembersihan data sebelum proses analisis lebih lanjut.

```
[14]: df_penjualan = df_penjualan.dropna()
print(df_penjualan.isnull().sum())
```

Tgl	0
Jenis	0
Kode	0
Produk	0
Deskripsi	0
Kuantitas	0
Satuan	0
HPP satuan	0
HPP	0
Harga satuan	0
Total Harga	0
Harga yang dibayarkan	0
Pembayaran	0
Tahun	0
Sheet	0
dtype:	int64

Gambar 3.17 Hasil Pengecekan Nilai Kosong Setelah Penghapusan Baris

Gambar 3.17 di atas menunjukkan hasil setelah dilakukan proses pembersihan data dengan fungsi `dropna()` pada *DataFrame* `df_penjualan`. Fungsi ini menghapus semua baris yang mengandung nilai kosong. Setelah proses ini dilakukan, hasil pemeriksaan ulang menggunakan `isnull().sum()` menunjukkan bahwa seluruh kolom dalam *DataFrame* tersebut sudah tidak memiliki nilai kosong, dengan total nol *missing values* pada setiap kolom. Langkah ini penting untuk memastikan integritas data sebelum dilakukan proses analisis atau visualisasi lebih lanjut.

```
[15]: print(df_pembelian.isnull().sum())
```

Tgl	3
Jenis	3
Kode	3
Produk	3
Deskripsi	3
Kuantitas	3
Satuan	3
Harga	3
Total Harga	3
Harga yang dibayarkan	3
Pembayaran	0
Tahun	0
Sheet	0
dtype:	int64

Gambar 3.18 Mengecek Jumlah Nilai Kosong

Gambar 3.18 diatas menunjukkan hasil pemeriksaan nilai kosong pada *DataFrame* *df_pembelian*. Terlihat bahwa terdapat 3 baris yang memiliki nilai kosong pada sejumlah kolom penting seperti Tgl, Jenis, Kode, Produk, Deskripsi, Kuantitas, Satuan, Harga, Total Harga, dan Harga yang dibayarkan. Sementara itu, kolom Pembayaran, Tahun, dan *Sheet* tidak mengandung nilai kosong. Identifikasi nilai kosong ini merupakan langkah awal yang penting dalam proses pembersihan data sebelum analisis lebih lanjut dapat dilakukan.

```
[16]: df_pembelian = df_pembelian.dropna()
      print(df_pembelian.isnull().sum())
```

Tgl	0
Jenis	0
Kode	0
Produk	0
Deskripsi	0
Kuantitas	0
Satuan	0
Harga	0
Total Harga	0
Harga yang dibayarkan	0
Pembayaran	0
Tahun	0
Sheet	0
dtype:	int64

Gambar 3.19 Hasil Pengecekan Nilai Kosong Setelah Penghapusan Baris

Gambar 3.19 diatas menunjukkan bahwa proses pembersihan data pada *DataFrame* *df_pembelian* telah berhasil dilakukan dengan menggunakan metode *dropna()*, yang menghapus baris-baris yang memiliki nilai kosong. Setelah pembersihan, seluruh kolom termasuk Tgl, Jenis, Kode, Produk, dan lainnya, tidak lagi mengandung nilai kosong. Hal ini memastikan bahwa data sudah bersih dan siap untuk dianalisis lebih lanjut tanpa gangguan dari data yang tidak lengkap.

```
[17]: jumlah_duplikat = df_pengeluaran.duplicated().sum()
      print(f"Jumlah data duplikat: {jumlah_duplikat}")
```

Jumlah data duplikat: 9

Gambar 3.20 Pengecekan Jumlah Data Duplikat pada *df_pengeluaran*

Gambar 3.20 diatas menunjukkan hasil dari proses pemeriksaan data duplikat pada *DataFrame* *df_pengeluaran*. Berdasarkan *output* yang

ditampilkan, terdapat 9 baris data yang teridentifikasi sebagai duplikat. Artinya, ada sembilan entri dalam data pengeluaran yang memiliki nilai identik di seluruh kolom dengan entri lainnya. Langkah ini penting dilakukan untuk memastikan kualitas data, karena data duplikat dapat menyebabkan analisis menjadi bias atau tidak akurat.

```
[18]: df_duplikat = df_pengeluaran[df_pengeluaran.duplicated()]
df_duplikat
```

```
[18]:
```

	Tgl	Metode Pembayaran	Jenis Pengeluaran	Deskripsi	Jumlah	Tahun	Sheet
62	2022-12-23	Mestika				2022	Pengeluaran
73	2023-01-24	Mestika				2023	Pengeluaran
79	2023-01-25	Mestika				2023	Pengeluaran
90	2023-01-30	Mestika				2023	Pengeluaran
395	2024-05-30	Uang Tunai				2024	Pengeluaran
600	2024-09-19	Uang Tunai				2024	Pengeluaran
675	2024-10-31	Uang Tunai				2024	Pengeluaran
807	2024-12-30	Uang Tunai				2024	Pengeluaran
816	2024-12-30	Uang Tunai				2024	Pengeluaran

Gambar 3.21 Baris Data Duplikat yang Ditemukan pada df_pengeluaran

Gambar 3.21 diatas data konfidensial yang menunjukkan hasil dari proses identifikasi data duplikat pada *DataFrame* df_pengeluaran menggunakan metode duplicated(). Data yang ditampilkan adalah 9 baris yang terdeteksi sebagai duplikat, yang berarti seluruh nilai di setiap kolom dalam baris-baris tersebut identik dengan baris lain di dataset.

Keberadaan duplikat seperti ini berisiko menyebabkan hasil analisis menjadi tidak akurat karena informasi yang dihitung akan terduplikasi. Oleh karena itu, penting untuk melakukan pembersihan data, misalnya dengan menghapus baris-baris duplikat menggunakan drop_duplicates().

```
[19]: df_pengeluaran = df_pengeluaran.drop_duplicates()

[20]: jumlah_duplikat = df_pengeluaran.duplicated().sum()
      print(f"Jumlah data duplikat: {jumlah_duplikat}")

      Jumlah data duplikat: 0
```

Gambar 3.22 Cek Jumlah Data Duplikat Setelah Penghapusan

Gambar 3.22 diatas menunjukkan bahwa data duplikat pada *DataFrame* *df_pengeluaran* telah dibersihkan menggunakan fungsi `drop_duplicates()`. Kemudian, dilakukan pengecekan ulang menggunakan `duplicated().sum()` untuk memastikan bahwa tidak ada lagi data yang duplikat. Artinya, seluruh baris yang sebelumnya teridentifikasi sebagai duplikat berhasil dihapus, dan sekarang dataset *df_pengeluaran* bersih dari duplikasi.

```
[21]: jumlah_duplikat = df_penjualan.duplicated().sum()
      print(f"Jumlah data duplikat: {jumlah_duplikat}")

      Jumlah data duplikat: 0
```

Gambar 3.23 Cek Jumlah Data Duplikat pada *df_penjualan*

Gambar 3.23 diatas menunjukkan hasil pengecekan duplikasi pada *DataFrame* *df_penjualan* menggunakan fungsi `duplicated().sum()`. Artinya, tidak terdapat baris yang sama di dalam dataset *df_penjualan*. Dengan demikian, data penjualan dinyatakan bersih dari duplikasi dan dapat langsung digunakan untuk proses analisis selanjutnya tanpa perlu dilakukan penghapusan duplikat. Ini menunjukkan bahwa proses pembersihan data telah berjalan dengan baik untuk aspek ini.

```
[22]: jumlah_duplikat = df_pembelian.duplicated().sum()
      print(f"Jumlah data duplikat: {jumlah_duplikat}")

      Jumlah data duplikat: 36
```

Gambar 3.24 Cek Jumlah Data Duplikat pada *df_pembelian*

Gambar 3.24 diatas menunjukkan hasil pengecekan duplikasi pada *DataFrame* *df_pembelian* menggunakan fungsi `duplicated().sum()`. Artinya,

terdapat 36 baris data yang sama di dalam dataset `df_pembelian`. Hal ini mengindikasikan bahwa masih terdapat data yang berulang dan perlu dilakukan pembersihan lebih lanjut, yaitu dengan menghapus baris-baris duplikat tersebut.

```
[23]: df_duplikat = df_pembelian[df_pembelian.duplicated()]
df_duplikat
```

	Tgl	Jenis	Kode	Produk	Deskripsi	Kuantitas	Satuan	Harga	Total Harga	Harga yang dibayarkan	Pembayaran	Tahun	Sheet
8	2022-08-08	Persediaan	LN	Lidi Nipah								2022	Pembelian
100	2022-11-27	Persediaan	LN	Lidi Nipah								2022	Pembelian
101	2022-11-28	Persediaan	LN	Lidi Nipah								2022	Pembelian
103	2022-11-28	Persediaan	LN	Lidi Nipah								2022	Pembelian
379	2023-07-03	Persediaan	LS	Lidi Sawit								2023	Pembelian
415	2023-11-01	Persediaan	LK	Lidi Kelapa								2023	Pembelian
420	2023-11-02	Persediaan	LK	Lidi Kelapa								2023	Pembelian
424	2023-11-02	Persediaan	LK	Lidi Kelapa								2023	Pembelian
428	2024-01-01	Persediaan	LB	Lidi bonggol								2024	Pembelian
440	2024-01-25	Persediaan	LB	Lidi bonggol								2024	Pembelian
522	2024-06-01	Persediaan	LB	Lidi bonggol								2024	Pembelian
523	2024-06-01	Persediaan	LB	Lidi bonggol								2024	Pembelian
524	2024-06-01	Persediaan	LB	Lidi bonggol								2024	Pembelian

Gambar 3.25 Baris Data Duplikat yang Ditemukan pada `df_pembelian`

Gambar 3.25 menunjukkan data konfidensial yang terdapat data duplikat. Keberadaan data duplikat dapat memengaruhi hasil analisis, seperti menyebabkan total pembelian menjadi lebih besar dari yang seharusnya. Oleh karena itu, penting untuk menghapus duplikat ini agar analisis data lebih akurat dan merepresentasikan kondisi sebenarnya. Dengan demikian, langkah selanjutnya adalah membersihkan data duplikat tersebut menggunakan metode seperti `drop_duplicates()` sebelum melanjutkan ke proses analisis atau visualisasi data.

```
[24]: df_pembelian = df_pembelian.drop_duplicates()

[25]: jumlah_duplikat = df_pembelian.duplicated().sum()
      print(f"Jumlah data duplikat: {jumlah_duplikat}")

Jumlah data duplikat: 0
```

Gambar 3.26 Cek Jumlah Data Duplikat Setelah Penghapusan

Gambar 3.26 diatas data konfidensial yang menunjukkan bahwa data duplikat pada *DataFrame* df_pembelian telah dibersihkan menggunakan fungsi drop_duplicates(). Kemudian, dilakukan pengecekan ulang menggunakan duplicated().sum() untuk memastikan bahwa tidak ada lagi data yang duplikat. Artinya, seluruh baris yang sebelumnya teridentifikasi sebagai duplikat berhasil dihapus, dan sekarang dataset df_pembelian bersih dari duplikasi.

Langkah ini penting untuk memastikan integritas data agar hasil analisis tidak bias akibat adanya data yang berulang. Setelah proses ini, data siap digunakan untuk analisis lebih lanjut dengan lebih akurat dan terpercaya.

3.2.4 Analisis Data Operasional

Setelah melalui tahap pembersihan dan penggabungan data yang teliti, fondasi untuk analisis yang mendalam telah berhasil dibangun. Kini, data yang sebelumnya mentah telah berubah menjadi satu set data yang bersih, terstruktur, dan siap untuk digunakan. Tahap selanjutnya bukanlah sekadar menjalankan kode secara acak, melainkan merumuskan sebuah strategi analisis yang terarah. Tujuan utamanya adalah untuk menganalisis data mengubah deretan angka dan teks menjadi gambaran yang dapat dipahami tentang kinerja operasional perusahaan selama tiga tahun terakhir.

Perencanaan analisis dimulai dengan mendefinisikan pertanyaan-pertanyaan bisnis kunci yang ingin dijawab. Pertanyaan pertama dan paling fundamental adalah, "Bagaimana kondisi keuangan inti perusahaan?" Untuk menjawab ini, dirancanglah serangkaian perhitungan metrik keuangan utama atau *Key Performance Indicators* (KPI). Rencana ini mencakup kalkulasi Total

Pendapatan untuk mengukur kekuatan penjualan, Total Pengeluaran untuk memonitor efisiensi biaya, yang kemudian akan bermuara pada perhitungan Laba Kotor dan Laba Bersih. Metrik-metrik ini akan menjadi landasan utama dari *dashboard* yang akan dibangun, memberikan gambaran mengenai profitabilitas perusahaan. Lebih jauh lagi, direncanakan juga perhitungan rasio seperti Margin Laba Bersih untuk memahami seberapa efisien perusahaan dalam mengubah pendapatan menjadi keuntungan nyata.

Namun, analisis tidak berhenti pada gambaran besar saja. Rencana diperluas untuk menggali lebih dalam ke aspek operasional yang spesifik. Dirancang sebuah analisis untuk mengukur pertumbuhan pendapatan dari tahun ke tahun (Year-over-Year Growth) guna melihat lintasan bisnis jangka panjang. Selain itu, direncanakan pula analisis mendalam terhadap struktur pengeluaran, dengan tujuan untuk mengidentifikasi kategori-kategori biaya terbesar yang menjadi beban utama perusahaan. Analisis tren bulanan juga dimasukkan dalam rencana untuk mengungkap pola-pola musiman yang mungkin terjadi. Terakhir, untuk memahami efisiensi rantai pasok, direncanakan perbandingan antara kuantitas barang yang dibeli dengan yang berhasil dijual.

Perencanaan ini memastikan bahwa proses coding yang akan dilakukan menjadi lebih terstruktur, efisien, dan fokus untuk menghasilkan wawasan yang benar-benar relevan dan berdampak bagi PT Djaya Universal Indonesia.

```
[26]: #Total Pendapatan
total_pendapatan_tahunan = df_penjualan.groupby('Tahun')['Total Harga'].sum().reset_index()
total_pendapatan_tahunan.rename(columns={'Total Harga': 'Total Pendapatan'}, inplace=True)
```

Gambar 3.27 Kode untuk Menghitung Total Pendapatan

Gambar 3.27 di atas menunjukkan kode yang digunakan untuk menghitung total pendapatan perusahaan per tahun berdasarkan data penjualan. Proses ini dilakukan dengan mengelompokkan data berdasarkan kolom Tahun dan menjumlahkan nilai dari kolom Total Harga untuk setiap tahun. Selanjutnya, hasil agregasi tersebut diubah namanya menjadi Total Pendapatan

agar lebih mudah dipahami. Analisis ini berguna untuk memberikan gambaran umum terhadap tren pendapatan perusahaan dari tahun 2022 hingga 2024.

```
[27]: #Total Pengeluaran
total_pengeluaran_tahunan = df_pengeluaran.groupby('Tahun')['Jumlah'].sum().reset_index()
total_pengeluaran_tahunan.rename(columns={'Jumlah': 'Total Pengeluaran'}, inplace=True)
```

Gambar 3.28 Kode untuk Menghitung Total Pengeluaran

Gambar 3.28 di atas menunjukkan kode yang digunakan untuk menghitung total pengeluaran perusahaan per tahun berdasarkan data pengeluaran. Data dikelompokkan berdasarkan kolom Tahun, kemudian nilai pada kolom Jumlah dijumlahkan untuk masing-masing tahun. Hasilnya kemudian diubah namanya menjadi Total Pengeluaran untuk mempermudah pemahaman saat dilakukan analisis lebih lanjut.

```
[28]: #Laba Kotor
df_penjualan.loc[:, 'Laba Kotor Per Transaksi'] = df_penjualan['Total Harga'] - df_penjualan['HPP']
laba_kotor_tahunan = df_penjualan.groupby('Tahun')['Laba Kotor Per Transaksi'].sum().reset_index()
laba_kotor_tahunan.rename(columns={'Laba Kotor Per Transaksi': 'Total Laba Kotor'}, inplace=True)

financial_summary = pd.merge(total_pendapatan_tahunan, laba_kotor_tahunan, on='Tahun', how='left')
financial_summary = pd.merge(financial_summary, total_pengeluaran_tahunan, on='Tahun', how='left')
```

Gambar 3.29 Kode untuk Menghitung Laba Kotor

Gambar 3.29 di atas menunjukkan kode yang digunakan untuk menghitung laba kotor perusahaan berdasarkan data penjualan. Laba kotor dihitung dengan mengurangkan nilai HPP (Harga Pokok Produksi) dari Total Harga untuk setiap transaksi, dan hasilnya disimpan dalam kolom baru Laba Kotor Per Transaksi. Selanjutnya, nilai laba kotor dijumlahkan per tahun, kemudian digabungkan dengan total pendapatan dan total pengeluaran tahunan untuk membentuk ringkasan keuangan (financial summary) yang dapat digunakan dalam analisis performa bisnis.

```
[29]: #Laba Bersih
financial_summary['Total Laba Bersih'] = financial_summary['Total Laba Kotor'] - financial_summary['Total Pengeluaran']
```

Gambar 3.30 Kode untuk Menghitung Laba Bersih

Gambar 3.30 di atas menunjukkan kode yang digunakan untuk menghitung Total Laba Bersih perusahaan. Perhitungan dilakukan dengan

mengurangkan Total Pengeluaran dari Total Laba Kotor yang telah diperoleh sebelumnya. Hasil perhitungan ini ditambahkan ke dalam kolom baru bernama Total Laba Bersih di dalam *DataFrame* `financial_summary`. Nilai ini menjadi indikator utama untuk mengetahui performa keuangan perusahaan setiap tahunnya.

```
[30]: #Margin Laba Kotor (%)
# Menggunakan .fillna(0) untuk menangani kasus pembagian dengan nol (jika pendapatan 0)
financial_summary['Margin Laba Kotor (%)'] = (financial_summary['Total Laba Kotor'] / financial_summary['Total Pendapatan'] * 100).fillna(0)
```

Gambar 3.31 Kode untuk Menghitung Margin Laba Kotor dalam Persen

Gambar 3.31 di atas menunjukkan kode yang digunakan untuk menghitung Margin Laba Kotor (%), yaitu persentase laba kotor terhadap total pendapatan perusahaan. Perhitungan dilakukan dengan membagi nilai Total Laba Kotor dengan Total Pendapatan, kemudian dikalikan 100 agar hasilnya dalam bentuk persentase. Fungsi `.fillna(0)` ditambahkan untuk menangani potensi pembagian dengan nol, sehingga hasil perhitungan tetap valid dan tidak menimbulkan error.

```
[31]: #Margin Laba Bersih (%)
financial_summary['Margin Laba Bersih (%)'] = (financial_summary['Total Laba Bersih'] / financial_summary['Total Pendapatan'] * 100).fillna(0)
```

Gambar 3.32 Kode untuk Menghitung Margin Laba Bersih dalam Persen

Gambar 3.32 di atas menunjukkan kode yang digunakan untuk menghitung Margin Laba Bersih (%), yaitu persentase laba bersih terhadap total pendapatan. Perhitungan dilakukan dengan membagi kolom Total Laba Bersih dengan Total Pendapatan, lalu dikalikan 100 untuk mendapatkan hasil dalam satuan persen. Fungsi `.fillna(0)` digunakan untuk mengantisipasi kemungkinan adanya nilai nol pada pendapatan yang dapat menyebabkan kesalahan pembagian, sehingga margin tetap dapat dihitung dengan aman.

```
[32]: #Pertumbuhan Pendapatan YoY (%)
financial_summary.sort_values(by='Tahun', inplace=True)
financial_summary['Pendapatan Tahun Lalu'] = financial_summary['Total Pendapatan'].shift(1)
financial_summary['Pertumbuhan Pendapatan YoY (%)'] = (
    (financial_summary['Total Pendapatan'] - financial_summary['Pendapatan Tahun Lalu']) / financial_summary['Pendapatan Tahun Lalu']) * 100).fillna(0)
financial_summary.drop(columns=['Pendapatan Tahun Lalu'], inplace=True)

print("Analisis Keuangan Utama (KPI per Tahun)\n")

pd.options.display.float_format = '{:,.2f}'.format
print(financial_summary[[
    'Tahun',
    'Total Pendapatan',
    'Total Laba Kotor',
    'Total Laba Bersih',
    'Margin Laba Kotor (%)',
    'Margin Laba Bersih (%)',
    'Pertumbuhan Pendapatan YoY (%)'
]].to_string(index=False))
pd.options.display.float_format = None
```

Analisis Keuangan Utama (KPI per Tahun)

Tahun	Total Pendapatan	Total Laba Kotor	Total Laba Bersih	Margin Laba Kotor (%)	Margin Laba Bersih (%)	Pertumbuhan Pendapatan YoY (%)
2022						
2023						
2024						

Gambar 3.33 Kode dan Hasil Perhitungan Analisis Keuangan Utama

Gambar 3.33 di atas menunjukkan kode yang digunakan untuk menghitung pertumbuhan pendapatan tahunan (YoY/Year-over-Year) dalam bentuk persentase. Perhitungan ini dilakukan dengan membandingkan total pendapatan suatu tahun dengan pendapatan tahun sebelumnya. Hasil perhitungan data konfidensial ditambahkan sebagai kolom baru bernama Pertumbuhan Pendapatan YoY (%).

Setelah seluruh indikator keuangan utama dihitung, data konfidensial yang ditampilkan dalam bentuk tabel yang berisi kolom-kolom seperti Total Pendapatan, Total Laba Kotor, Total Laba Bersih, Margin Laba Kotor, Margin Laba Bersih, dan Pertumbuhan Pendapatan YoY. Tabel ini memberikan ringkasan performa keuangan perusahaan dari tahun 2022 hingga 2024.

```
[33]: print("Total Pengeluaran per Tahun\n")
total_pengeluaran_tahunan['Tahun'] = total_pengeluaran_tahunan['Tahun'].astype(int)

pd.options.display.float_format = 'Rp{:, .2f}'.format
print(total_pengeluaran_tahunan.to_string(index=False))
pd.options.display.float_format = None
```

Total Pengeluaran per Tahun

Tahun	Total Pengeluaran
2022	
2023	
2024	

Gambar 3.34 Kode dan Hasil Total Pengeluaran per Tahun

memberikan pandangan mendetail tentang alokasi biaya perusahaan untuk setiap kategori pengeluaran dari tahun 2022 hingga 2024.

```
[35]: print("Perbandingan Kuantitas Pembelian vs Penjualan per Produk\n")
kuantitas_pembelian_produk = df_pembelian.groupby('Produk')['Kuantitas'].sum().reset_index()
kuantitas_pembelian_produk.rename(columns={'Kuantitas': 'Kuantitas Dibeli'}, inplace=True)

kuantitas_penjualan_produk = df_penjualan.groupby('Produk')['Kuantitas'].sum().reset_index()
kuantitas_penjualan_produk.rename(columns={'Kuantitas': 'Kuantitas Terjual'}, inplace=True)

perbandingan_kuantitas = pd.merge(
    kuantitas_pembelian_produk,
    kuantitas_penjualan_produk,
    on='Produk',
    how='outer'
).fillna(0)

perbandingan_kuantitas['Selisih Kuantitas (Dibeli - Terjual)'] = perbandingan_kuantitas['Kuantitas Dibeli'] - perbandingan_kuantitas['Kuantitas Terjual']
perbandingan_kuantitas.sort_values(by='Kuantitas Terjual', ascending=False, inplace=True)

pd.options.display.float_format = '{:,.4f}'.format
print(perbandingan_kuantitas.head(5).to_string(index=False))
pd.options.display.float_format = None
```

Produk	Kuantitas Dibeli	Kuantitas Terjual	Selisih Kuantitas (Dibeli - Terjual)
Lidi bonggol			
Lidi Sawit			
Lidi potong			
Lidi kelapa			
Lidi Nipah			

Gambar 3.36 Kode dan Hasil Perbandingan Kuantitas Pembelian vs Penjualan

Gambar 3.36 di atas menunjukkan kode yang digunakan untuk membandingkan jumlah kuantitas produk yang dibeli dengan yang terjual. Proses ini dimulai dengan menghitung total kuantitas untuk setiap produk dari data pembelian dan data penjualan secara terpisah. Data tersebut kemudian digabungkan berdasarkan nama produk untuk menyatukannya dalam satu tabel. Selanjutnya, kode ini membuat kolom baru bernama Selisih Kuantitas (Dibeli - Terjual) yang berisi hasil pengurangan antara jumlah yang dibeli dan jumlah yang terjual. Analisis ini sangat berguna untuk manajemen inventaris, karena dapat menunjukkan produk mana yang mengalami surplus (lebih banyak dibeli daripada dijual) atau defisit (lebih banyak dijual daripada dibeli).

```
[36]: print("Rata-rata Nilai Pengeluaran per Transaksi per Tahun\n")
rata_rata_pengeluaran_per_transaksi = df_pengeluaran.groupby('Tahun')['Jumlah'].mean().reset_index()
rata_rata_pengeluaran_per_transaksi.rename(columns={'Jumlah': 'Rata-rata Nilai Pengeluaran'}, inplace=True)

pd.options.display.float_format = 'Rp{:, .2f}'.format
print(rata_rata_pengeluaran_per_transaksi.to_string(index=False))
pd.options.display.float_format = None

Rata-rata Nilai Pengeluaran per Transaksi per Tahun

Tahun  Rata-rata Nilai Pengeluaran
2022
2023
2024
```

Gambar 3.37 Kode dan Hasil Rata-rata Nilai Pengeluaran per Transaksi

Gambar 3.37 di atas menunjukkan kode yang digunakan untuk menghitung rata-rata nilai pengeluaran per transaksi untuk setiap tahunnya. Proses ini dilakukan dengan mengelompokkan data konfidensial pengeluaran (df_pengeluaran) berdasarkan kolom Tahun. Kemudian, untuk setiap tahun, dihitung nilai rata-rata dari kolom Jumlah. Nama kolom hasil perhitungan diubah menjadi Rata-rata Nilai Pengeluaran. Hasilnya kemudian diformat ke dalam mata uang Rupiah (Rp) dan ditampilkan dalam tabel, memberikan wawasan tentang rata-rata besaran transaksi pengeluaran dari tahun 2022 hingga 2024.

```
[37]: print("Pengeluaran Top 5 Deskripsi\n")
top_deskripsi_pengeluaran = df_pengeluaran.groupby('Jenis Pengeluaran')['Jumlah'].sum().reset_index()
top_deskripsi_pengeluaran.sort_values(by='Jumlah', ascending=False, inplace=True)

pd.options.display.float_format = 'Rp{:, .2f}'.format
print(top_deskripsi_pengeluaran.head(5).to_string(index=False))
pd.options.display.float_format = None

Pengeluaran Top 5 Deskripsi

Jenis Pengeluaran  Jumlah
```

Gambar 3.38 Kode dan Hasil 5 Deskripsi Pengeluaran Teratas

Gambar 3.38 di atas menunjukkan kode yang digunakan untuk mengidentifikasi 5 kategori pengeluaran terbesar perusahaan secara keseluruhan. Proses ini dilakukan dengan mengelompokkan seluruh data konfidensial pengeluaran berdasarkan kolom Jenis Pengeluaran dan menjumlahkan total pengeluaran untuk setiap kategori. Hasilnya kemudian diurutkan dari nilai terbesar ke terkecil, dan fungsi .head(5) digunakan untuk

mengambil hanya lima baris teratas. Tabel yang ditampilkan, setelah diformat ke dalam mata uang Rupiah (Rp), menyajikan daftar lima kategori dengan total pengeluaran tertinggi, yang berguna untuk mengetahui pos biaya utama dalam operasional perusahaan.

```
[38]: print("Rasio Pengeluaran terhadap Pendapatan per Tahun (%)\n")
      rasio_pengeluaran_pendapatan = financial_summary[['Tahun', 'Total Pendapatan', 'Total Pengeluaran']].copy()
      rasio_pengeluaran_pendapatan['Rasio Pengeluaran/Pendapatan (%)'] = (rasio_pengeluaran_pendapatan['Total Pengeluaran'] /
      rasio_pengeluaran_pendapatan['Total Pendapatan'] * 100).fillna(0)

      pd.options.display.float_format = '{:,.2f}%'.format
      print(rasio_pengeluaran_pendapatan[['Tahun', 'Rasio Pengeluaran/Pendapatan (%)']].to_string(index=False))
      pd.options.display.float_format = None

Rasio Pengeluaran terhadap Pendapatan per Tahun (%)
Tahun Rasio Pengeluaran/Pendapatan (%)
2022
2023
2024
```

Gambar 3.39 Kode dan Hasil Rasio Pengeluaran terhadap Pendapatan

Gambar 3.39 di atas menunjukkan kode yang digunakan untuk menghitung rasio data konfidensial pengeluaran terhadap pendapatan tahunan dalam bentuk persentase. Proses ini dilakukan dengan membagi nilai Total Pengeluaran dengan Total Pendapatan untuk setiap tahun, kemudian hasilnya dikalikan 100. Fungsi `.fillna(0)` ditambahkan untuk memastikan tidak ada eror jika terjadi pembagian dengan nol. Rasio ini merupakan indikator penting untuk mengukur efisiensi operasional, di mana hasil akhirnya ditampilkan dalam tabel yang menunjukkan persentase pengeluaran dari total pendapatan perusahaan setiap tahunnya.

```
[39]: df_pembelian['Bulan'] = df_pembelian['Tgl'].dt.month
      df_penjualan['Bulan'] = df_penjualan['Tgl'].dt.month
      df_pengeluaran['Bulan'] = df_pengeluaran['Tgl'].dt.month
```

Gambar 3.40 Kode untuk Menambahkan Kolom Bulan

Gambar 3.40 di atas menunjukkan kode yang digunakan untuk membuat kolom baru bernama Bulan pada tiga *DataFrame* yang berbeda: `df_pembelian`, `df_penjualan`, dan `df_pengeluaran`. Untuk setiap *DataFrame*, kode ini mengekstrak nomor bulan (contohnya, 1 untuk Januari, 2 untuk Februari) dari kolom tanggal yang ada, yaitu `Tgl`. Proses ini menggunakan aksesor `.dt.month` yang berfungsi pada data bertipe `datetime`. Penambahan kolom Bulan ini

merupakan langkah persiapan data yang penting untuk memungkinkan analisis data berdasarkan periode bulanan.

```
[40]: print("Tren Kuantitas Penjualan per Bulan (Agregasi Seluruh Tahun)\n")
tren_kuantitas_penjualan_bulanan = df_penjualan.groupby('Bulan')['Kuantitas'].sum().reset_index()
tren_kuantitas_penjualan_bulanan['Nama Bulan'] = tren_kuantitas_penjualan_bulanan['Bulan'].apply(
    lambda x: pd.to_datetime(str(x), format='%m').strftime('%B'))

pd.options.display.float_format = '{:,.2f}'.format
print(tren_kuantitas_penjualan_bulanan[['Nama Bulan', 'Kuantitas']].to_string(index=False))
pd.options.display.float_format = None
```

Tren Kuantitas Penjualan per Bulan (Agregasi Seluruh Tahun)

Nama Bulan	Kuantitas
January	
February	
March	
April	
May	
June	
July	
August	
September	
October	
November	
December	

Gambar 3.41 Kode dan Hasil Tren Kuantitas Penjualan per Bulan

Gambar 3.41 di atas menunjukkan kode yang digunakan untuk menganalisis tren atau pola musiman data konfidensial penjualan dengan menghitung total kuantitas penjualan untuk setiap bulan. Data penjualan dari seluruh tahun digabungkan. Prosesnya dimulai dengan mengelompokkan data berdasarkan kolom Bulan, lalu menjumlahkan Kuantitas untuk setiap bulannya. Selanjutnya, sebuah kolom baru bernama Nama Bulan dibuat untuk mengubah format angka bulan (misalnya 1) menjadi nama bulan secara lengkap (misalnya Januari) agar laporan lebih mudah dibaca. Tabel hasil akhir menunjukkan total kuantitas penjualan kumulatif untuk setiap bulan, yang membantu dalam mengidentifikasi bulan-bulan dengan aktivitas penjualan tertinggi dan terendah.



```
[42]: print(" Analisis Harga Satuan Pembelian vs Penjualan per Produk\n")
avg_harga_beli_produk = df_pembelian.groupby('Produk')['Harga'].mean().reset_index()
avg_harga_beli_produk.rename(columns={'Harga': 'Rata-rata Harga Beli'}, inplace=True)

avg_harga_jual_produk = df_penjualan.groupby('Produk')['Harga satuan'].mean().reset_index()
avg_harga_jual_produk.rename(columns={'Harga satuan': 'Rata-rata Harga Jual'}, inplace=True)

perbandingan_harga_satuan = pd.merge(
    avg_harga_beli_produk,
    avg_harga_jual_produk,
    on='Produk',
    how='outer',
).fillna(0)

perbandingan_harga_satuan = pd.merge(
    perbandingan_harga_satuan,
    kuantitas_penjualan_produk,
    on='Produk',
    how='left',
).fillna(0)

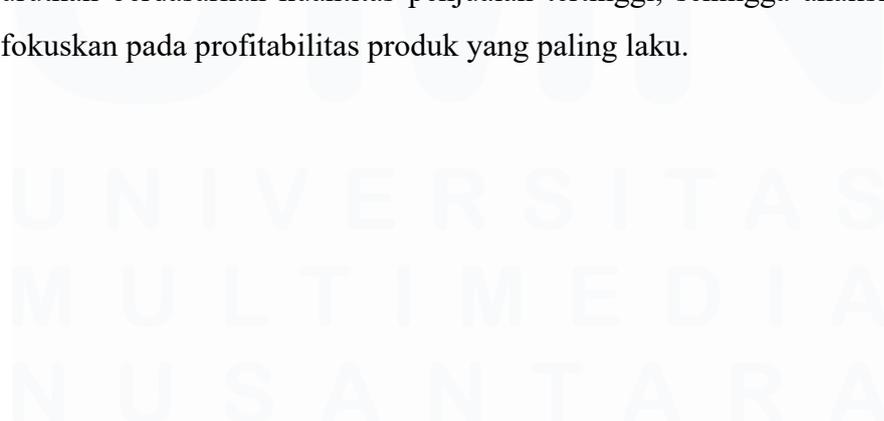
perbandingan_harga_satuan['Selisih Harga (Jual - Beli)'] = perbandingan_harga_satuan['Rata-rata Harga Jual'] - perbandingan_harga_satuan['Rata-rata Harga
perbandingan_harga_satuan['Margin Satuan (%)'] = ((perbandingan_harga_satuan['Selisih Harga (Jual - Beli)'] /
perbandingan_harga_satuan['Rata-rata Harga Beli']) * 100).fillna(0)
perbandingan_harga_satuan.sort_values(by='Kuantitas Terjual', ascending=False, inplace=True)

pd.options.display.float_format = '{:,.2f}'.format
print(perbandingan_harga_satuan.head(5).to_string(index=False))
pd.options.display.float_format = None
```

Produk	Rata-rata Harga Beli	Rata-rata Harga Jual	Kuantitas Terjual	Selisih Harga (Jual - Beli)	Margin Satuan (%)
Lidi bonggol					
Lidi Sawit					
Lidi potong					
Lidi kelapa					
Lidi Nipah					

Gambar 3.42 Kode dan Hasil Analisis Harga Satuan Pembelian vs Penjualan

Gambar 3.42 di atas menunjukkan kode yang digunakan untuk melakukan analisis data konfidensial profitabilitas per produk dengan membandingkan harga beli rata-rata dengan harga jual rata-rata. Proses ini diawali dengan menghitung harga beli rata-rata dari data pembelian dan harga jual rata-rata dari data penjualan untuk setiap produk. Kedua hasil tersebut kemudian digabungkan, bersama dengan data total kuantitas terjual, ke dalam satu tabel. Selanjutnya, kode ini menghitung dua metrik baru: Selisih Harga (Jual - Beli) untuk mengetahui laba kotor per unit, dan Margin Satuan (%) untuk mengetahui persentase keuntungan yang diambil dari harga beli. Tabel akhir diurutkan berdasarkan kuantitas penjualan tertinggi, sehingga analisis dapat difokuskan pada profitabilitas produk yang paling laku.



```
[43]: df_pengeluaran
```

[43]:		Tgl	Metode Pembayaran	Jenis Pengeluaran	Deskripsi	Jumlah	Tahun	Sheet	Bulan
	1	2022-09-02	Mestika				2022	Pengeluaran	9
	2	2022-09-05	Mestika				2022	Pengeluaran	9
	3	2022-09-17	Mestika				2022	Pengeluaran	9
	4	2022-09-18	Mestika				2022	Pengeluaran	9
	5	2022-09-18	Mestika				2022	Pengeluaran	9

	815	2024-12-30	Uang Tunai				2024	Pengeluaran	12
	817	2024-12-30	Uang Tunai				2024	Pengeluaran	12
	818	2024-12-30	Uang Tunai				2024	Pengeluaran	12
	819	2024-12-30	Uang Tunai				2024	Pengeluaran	12
	820	2024-12-30	Uang Tunai				2024	Pengeluaran	12

809 rows × 8 columns

Gambar 3.43 Hasil Akhir DataFrame df_pengeluaran

Gambar 3.43 di atas menampilkan isi dari *DataFrame* df_pengeluaran setelah melalui beberapa tahap pemrosesan sebelumnya. *DataFrame* ini berisi gabungan seluruh data konfidensial transaksi pengeluaran dari tahun 2022 hingga 2024, yang kini terdiri dari 809 baris data dan 8 kolom. Kolom-kolom yang ada memberikan rincian untuk setiap transaksi, mencakup tanggal, metode pembayaran, jenis pengeluaran, deskripsi, jumlah, tahun, *sheet* dan bulan. Tampilan ini mengonfirmasi bahwa data pengeluaran telah berhasil dikonsolidasikan dan terstruktur, sehingga siap untuk digunakan dalam analisis keuangan lebih lanjut.

```
[44]: df_penjualan
```

```
[44]:
```

	Tgl	Jenis	Kode	Produk	Deskripsi	Kuantitas	Satuan	HPP satuan	HPP	Harga satuan	Total Harga	Harga yang dibayarkan	Pembayaran	Tahun
1	2022-09-21	Persediaan	LN	Lidi Nipah	Penjualan Lidi nipah									2022
2	2022-10-31	Persediaan	LN	Lidi Nipah	Penjualan Lidi nipah									2022
3	2022-11-16	Persediaan	LS	Lidi Sawit	Penjualan Lidi sawit									2022
4	2022-11-23	Persediaan	LS	Lidi Sawit	Penjualan Lidi sawit									2022
5	2022-11-23	Persediaan	LS	Lidi Sawit	Penjualan Lidi sawit									2022
...
90	2024-12-09	Persediaan	LB	Lidi bonggol	Penjualan lidi bonggol									2024
91	2024-12-12	Persediaan	LB	Lidi bonggol	Penjualan lidi bonggol									2024
92	2024-12-12	Persediaan	LP	Lidi potong	Penjualan lidi potong									2024
93	2024-12-23	Persediaan	LB	Lidi bonggol	Penjualan lidi bonggol									2024
94	2024-12-23	Persediaan	LB	Lidi bonggol	Penjualan lidi bonggol									2024

92 rows x 17 columns

Gambar 3.44 Hasil Akhir DataFrame df_penjualan

Gambar 3.44 di atas menampilkan isi dari *DataFrame* df_penjualan setelah melalui proses penggabungan data konfidensial dari tahun 2022 hingga 2024. *DataFrame* ini terdiri dari 92 baris dan 17 kolom, di mana setiap baris merepresentasikan satu transaksi penjualan yang tercatat. Kolom-kolom yang tersedia memberikan informasi lengkap tentang setiap penjualan, mencakup tanggal, jenis, kode, produk, kuantitas, Harga Pokok Penjualan satuan (HPP), HPP, dan lainnya. Tampilan ini menunjukkan bahwa data penjualan dari seluruh periode telah berhasil disatukan dan siap menjadi dasar untuk analisis performa penjualan dan pendapatan perusahaan.

[45]: df_pembelian

[45]:

	Tgl	Jenis	Kode	Produk	Deskripsi	Kuantitas	Satuan	Harga	Total Harga	Harga yang dibayarkan	Pembayaran	Tahun	Sheet	Bulan
1	2022-07-17	Persediaan	LN	Lidi Nipah								2022	Pembelian	7
2	2022-07-18	Persediaan	LN	Lidi Nipah								2022	Pembelian	7
3	2022-07-18	Persediaan	LN	Lidi Nipah								2022	Pembelian	7
4	2022-07-20	Persediaan	LN	Lidi Nipah								2022	Pembelian	7
5	2022-07-21	Persediaan	LN	Lidi Nipah								2022	Pembelian	7
...
860	2024-12-29	Persediaan	LP	Lidi potong								2024	Pembelian	12
861	2024-12-31	Persediaan	LP	Lidi potong								2024	Pembelian	12
862	2024-12-31	Persediaan	LP	Lidi potong								2024	Pembelian	12
863	2024-12-31	Persediaan	LP	Lidi potong								2024	Pembelian	12
864	2024-12-31	Persediaan	LK	Lidi kelapa								2024	Pembelian	12

826 rows x 14 columns

Gambar 3.45 Hasil Akhir DataFrame df_pembelian

Gambar 3.45 di atas menampilkan isi dari *DataFrame* df_pembelian setelah melalui proses penggabungan dan pembersihan data. *DataFrame* ini kini berisi catatan lengkap dari seluruh data konfidensial transaksi pembelian yang dilakukan perusahaan dari tahun 2022 hingga 2024, yang terdiri dari 826 baris dan 14 kolom. Setiap baris menyajikan rincian satu transaksi pembelian, termasuk informasi tanggal, jenis, kode, produk, deskripsi, kuantitas, dan lainnya. Tabel yang telah terkonsolidasi ini merupakan dasar yang fundamental untuk melakukan analisis biaya pengadaan, manajemen inventaris, dan aktivitas lain yang berkaitan dengan pembelian.

```
[46]: with pd.ExcelWriter('data_bersih.xlsx', engine='openpyxl') as writer:
df_pembelian.to_excel(writer, sheet_name='Pembelian', index=False)
df_penjualan.to_excel(writer, sheet_name='Penjualan', index=False)
df_pengeluaran.to_excel(writer, sheet_name='Pengeluaran', index=False)
```

Gambar 3.46 Kode untuk Menyimpan DataFrame

Gambar 3.46 di atas menunjukkan kode yang berfungsi untuk mengekspor atau menyimpan tiga *DataFrame* utama yang telah diproses (df_pembelian, df_penjualan, dan df_pengeluaran) ke dalam satu *file* Excel baru. Dengan menggunakan pd.ExcelWriter, sebuah *file* bernama data_bersih.xlsx dibuat. Kemudian, setiap *DataFrame* ditulis ke dalam *sheet* yang berbeda di dalam *file*

tersebut: `df_pembelian` ke *sheet* Pembelian, `df_penjualan` ke Penjualan, dan `df_pengeluaran` ke Pengeluaran. Argumen `index=False` digunakan untuk mencegah penulisan kolom indeks *DataFrame* ke dalam *file* Excel, sehingga menghasilkan output yang lebih bersih.

3.2.5 Visualisasi Data dan Pengembangan Dashboard

Setelah melalui tahap pembersihan, integrasi, dan analisis data yang komprehensif menggunakan *Python*, alur kerja memasuki tahap yang visualisasi data dan pengembangan *dashboard* interaktif. Pada tahap ini, data yang sebelumnya telah diproses dan dipastikan bersih, nilai kosong, maupun duplikat, kini siap untuk diubah menjadi wawasan visual yang berdampak.

Langkah pertama adalah mengimpor set data yang telah terstruktur tersebut, yang telah diekspor ke dalam satu *file* Excel bernama `data_bersih.xlsx`, ke dalam platform intelijen bisnis Microsoft Power BI. Tujuan utama dari tahap ini adalah untuk menerjemahkan deretan data dan angka menjadi sebuah laporan visual yang dinamis dan mudah dipahami, sehingga dapat mendukung manajemen dalam melakukan eksplorasi data dan pengambilan keputusan strategis yang berbasis bukti. Untuk mencapai hal tersebut, fondasi analitis di dalam *dashboard* perlu dibangun terlebih dahulu. Langkah krusial untuk membangun fondasi ini adalah dengan merumuskan serangkaian metrik atau *measures* menggunakan bahasa DAX (*Data Analysis Expressions*). *Measures* ini berfungsi sebagai mesin kalkulasi yang akan menggerakkan semua visualisasi pada *dashboard*, menghitung berbagai Indikator Kinerja Utama (KPI) secara otomatis. Berikut adalah rincian *measures* utama yang dikembangkan untuk *dashboard* PT Djaya Universal Indonesia.



```
1 Total Pendapatan = SUM(Penjualan[Total Harga])
```

Gambar 3.47 Rumus DAX untuk Menghitung Total Pendapatan

Gambar 3.47 di atas menunjukkan *measure* yang digunakan untuk menghitung Total Pendapatan. Formula ini bekerja dengan cara menjumlahkan (SUM) semua nilai dari kolom [Total Harga] yang ada di dalam tabel Penjualan. Hasilnya adalah metrik keuangan utama yang menunjukkan performa penjualan perusahaan secara keseluruhan

```
1 Laba Kotor = SUM('Penjualan'[Total Harga]) - SUM('Penjualan'[HPP])
```

Gambar 3.48 Rumus DAX untuk Menghitung Laba Kotor

Gambar 3.48 di atas menunjukkan *measure* untuk menghitung Laba Kotor. Metrik ini digunakan untuk mengukur profitabilitas inti perusahaan dari aktivitas penjualan sebelum dikurangi biaya operasional. Formula ini bekerja dengan menghitung selisih antara total pendapatan dengan total Harga Pokok Penjualan (HPP).

```
1 Laba Bersih = [Laba Kotor] - [Total Pengeluaran]
```

Gambar 3.49 Rumus DAX untuk Menghitung Laba Bersih

Gambar 3.49 di atas menampilkan *measure* yang menghitung Laba Bersih, yang merupakan indikator profitabilitas akhir perusahaan setelah memperhitungkan seluruh pendapatan dan pengeluaran. Formula ini bekerja dengan cara sederhana, yaitu mengurangi *measure* [Total Pengeluaran] dari *measure* [Laba Kotor] yang telah dihitung sebelumnya. Metrik ini menjadi tolok ukur untuk melihat finansial perusahaan secara keseluruhan, karena menunjukkan sisa keuntungan yang sebenarnya setelah semua biaya operasional ditanggung.

```
1 Margin Laba Kotor (%) = DIVIDE([Laba Kotor], [Total Pendapatan])
```

Gambar 3.50 Rumus DAX untuk Menghitung Margin Laba Kotor (%)

Gambar 3.50 di atas menunjukkan *measure* yang digunakan untuk menghitung Margin Laba Kotor (%). Metrik ini untuk mengukur tingkat efisiensi perusahaan dalam menghasilkan laba dari setiap rupiah pendapatan

yang diterima, sebelum dipotong oleh biaya-biaya operasional. Formula ini bekerja dengan membagi nilai dari *measure* [Laba Kotor] dengan nilai *measure* [Total Pendapatan].

```
1 Margin Laba Bersih (%) = DIVIDE([Laba Bersih], [Total Pendapatan])
```

Gambar 3.51 Rumus DAX untuk Menghitung Margin Laba Bersih (%)

Gambar 3.51 di atas menampilkan *measure* Margin Laba Bersih (%), yang merupakan indikator kunci untuk menilai profitabilitas akhir dan efisiensi perusahaan secara keseluruhan. Metrik ini menunjukkan persentase keuntungan bersih yang tersisa dari setiap rupiah pendapatan setelah semua biaya, baik Harga Pokok Penjualan (HPP) maupun biaya operasional, telah dikurangi. Formula ini bekerja dengan membagi *measure* [Laba Bersih] dengan *measure* [Total Pendapatan].

```
1 Pertumbuhan Pendapatan YoY =  
2 VAR CurrentYearRevenue = [Total Pendapatan]  
3 VAR PreviousYearRevenue = CALCULATE(  
4     [Total Pendapatan],  
5     SAMEPERIODLASTYEAR('Penjualan'[Tgl])  
6 )  
7 RETURN  
8     DIVIDE(  
9         CurrentYearRevenue - PreviousYearRevenue, PreviousYearRevenue, BLANK()  
10    )
```

Gambar 3.52 Rumus DAX untuk Menghitung Pertumbuhan Pendapatan YoY

Gambar 3.52 di atas menampilkan *measure* Pertumbuhan Pendapatan YoY, yang dirancang untuk mengukur laju pertumbuhan pendapatan perusahaan dari tahun ke tahun. Formula ini pertama-tama menyimpan pendapatan tahun berjalan, lalu menggunakan fungsi SAMEPERIODLASTYEAR untuk mencari dan menyimpan pendapatan pada periode yang sama di tahun sebelumnya. Kemudian, *measure* ini menghitung selisih antara keduanya dan membaginya dengan pendapatan tahun lalu untuk menghasilkan persentase pertumbuhan. Metrik ini memberikan wawasan

krusial mengenai lintasan dan akselerasi bisnis perusahaan dalam jangka panjang.

```
1 Rata-rata Nilai Transaksi = DIVIDE([Total Pendapatan], [Jumlah Transaksi Penjualan], BLANK())
```

Gambar 3.53 Rumus DAX untuk Menghitung Rata-rata Nilai Transaksi

Gambar 3.53 di atas menampilkan *measure* Rata-rata Nilai Transaksi, yang bertujuan untuk menghitung nilai rata-rata dari setiap transaksi penjualan yang dilakukan perusahaan. Formula ini bekerja dengan membagi total pendapatan, yang diwakili oleh *measure* [Total Pendapatan], dengan jumlah total transaksi, yang diwakili oleh *measure* [Jumlah Transaksi Penjualan].

```
1 Jumlah Transaksi Penjualan = COUNTROWS('Penjualan')
```

Gambar 3.54 Rumus DAX untuk Menghitung Jumlah Transaksi Penjualan

Gambar 3.54 di atas menampilkan *measure* dasar Jumlah Transaksi Penjualan. Metrik ini berfungsi untuk menghitung total banyaknya transaksi penjualan yang telah dilakukan oleh perusahaan. Perhitungan ini dilakukan secara efisien dengan menggunakan fungsi COUNTROWS, yang secara langsung menghitung jumlah seluruh baris yang ada di dalam tabel 'Penjualan'.

```
1 Total Pengeluaran = SUM(Pengeluaran[Jumlah])
```

Gambar 3.55 Rumus DAX untuk Menghitung Total Pengeluaran

Gambar 3.55 di atas menampilkan *measure* Total Pengeluaran, sebuah metrik fundamental yang berfungsi untuk menghitung total akumulasi dari seluruh biaya operasional yang dikeluarkan oleh perusahaan. Formula ini bekerja dengan menjumlahkan (SUM) semua nilai yang tercatat pada kolom [Jumlah] di dalam tabel Pengeluaran.

```
1 Persentase Pengeluaran vs Pendapatan = DIVIDE([Total Pengeluaran], [Total Pendapatan], BLANK())
```

Gambar 3.56 Rumus DAX untuk Menghitung Pengeluaran vs Pendapatan

Gambar 3.56 di atas menampilkan *measure* Persentase Pengeluaran vs Pendapatan, yang berfungsi sebagai indikator penting untuk mengukur tingkat efisiensi operasional perusahaan. Formula ini menghitung rasio biaya dengan membagi total biaya operasional, yang diwakili oleh *measure* [Total Pengeluaran], dengan total pendapatan yang dihasilkan, yaitu *measure* [Total Pendapatan].

```
1 Rata-rata Nilai Pembelian = DIVIDE([Total Pembelian], [Jumlah Transaksi Pembelian], BLANK())
```

Gambar 3.57 Rumus DAX untuk Menghitung Rata-rata Nilai Pembelian

Gambar 3.57 di atas menampilkan *measure* Rata-rata Nilai Pembelian, yang berfungsi untuk menghitung nilai rata-rata dari setiap transaksi pembelian bahan baku yang dilakukan oleh perusahaan. Formula ini bekerja dengan membagi total nilai pembelian, yang diwakili oleh *measure* [Total Pembelian], dengan jumlah total transaksi pembelian, yaitu *measure* [Jumlah Transaksi Pembelian].

```
1 Total Pembelian = SUM('Pembelian'[Total Harga])
```

Gambar 3.58 Rumus DAX untuk Menghitung Total Pembelian

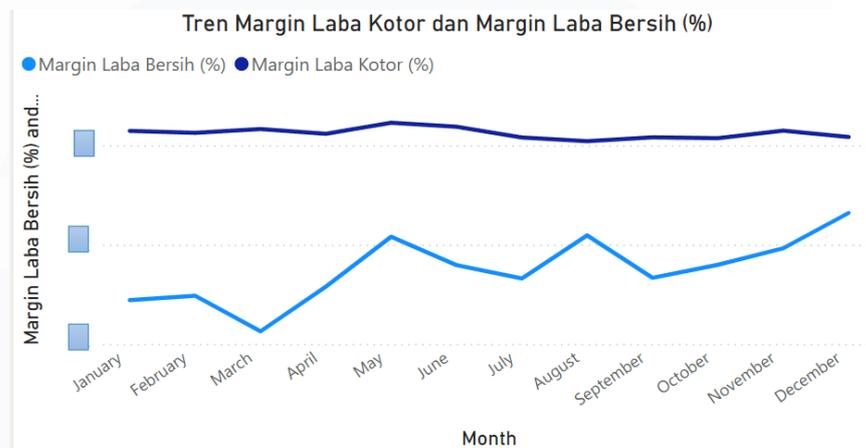
Gambar 3.58 di atas menampilkan *measure* dasar Total Pembelian. Metrik ini bertujuan untuk menghitung total nilai dari seluruh transaksi pembelian bahan baku yang dilakukan oleh perusahaan. Formula ini bekerja dengan cara menjumlahkan semua nilai yang tercatat pada kolom [Total Harga] dari tabel Pembelian.

```
1 Jumlah Transaksi Pembelian = COUNTROWS('Pembelian')
```

Gambar 3.59 Rumus DAX untuk Menghitung Jumlah Transaksi Pembelian

Gambar 3.59 di atas menampilkan *measure* dasar Jumlah Transaksi Pembelian, yang berfungsi untuk menghitung total frekuensi atau banyaknya transaksi pembelian yang dilakukan oleh perusahaan. Formula ini dilakukan

menggunakan fungsi COUNTROWS yang secara efisien menghitung jumlah seluruh baris yang terdapat di dalam tabel Pembelian.



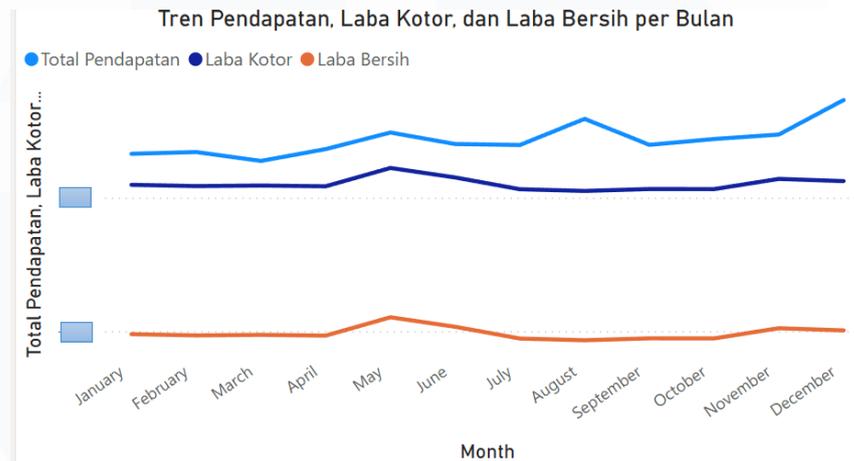
Gambar 3.60 Grafik Tren Margin Laba Kotor dan Margin Laba Bersih

Gambar 3.60 di atas menampilkan visualisasi tren bulanan yang membandingkan Margin Laba Kotor (%) dengan Margin Laba Bersih (%). Grafik ini secara efektif menyoroti sebuah performa finansial ganda, di mana terdapat kekuatan pada level operasional inti namun terdapat tantangan besar pada profitabilitas akhir.

Garis biru tua yang mewakili Margin Laba Kotor menunjukkan tren yang sangat stabil dan konsisten. Hal ini membuktikan bahwa fundamental bisnis perusahaan dalam menghasilkan produk dan menjualnya di atas harga pokok penjualan (HPP) berada dalam kondisi yang sangat solid dan sehat.

Sebaliknya, garis biru muda yang mewakili Margin Laba Bersih, secara konsisten berada di wilayah negatif sepanjang tahun. Ini menandakan bahwa perusahaan secara frekuen beroperasi dalam kondisi defisit setelah semua biaya operasional diperhitungkan. Kesenjangan yang signifikan antara kedua margin ini secara gamblang mengindikasikan bahwa laba kotor yang solid tersebut hampir seluruhnya terkikis oleh tingginya beban biaya operasional, yang menunjukkan adanya struktur biaya yang tidak efisien. Meskipun demikian, dapat diamati adanya tren perbaikan pada kuartal terakhir, di mana kurva

margin laba bersih bergerak naik, yang dapat mengisyaratkan adanya potensi perbaikan efisiensi.



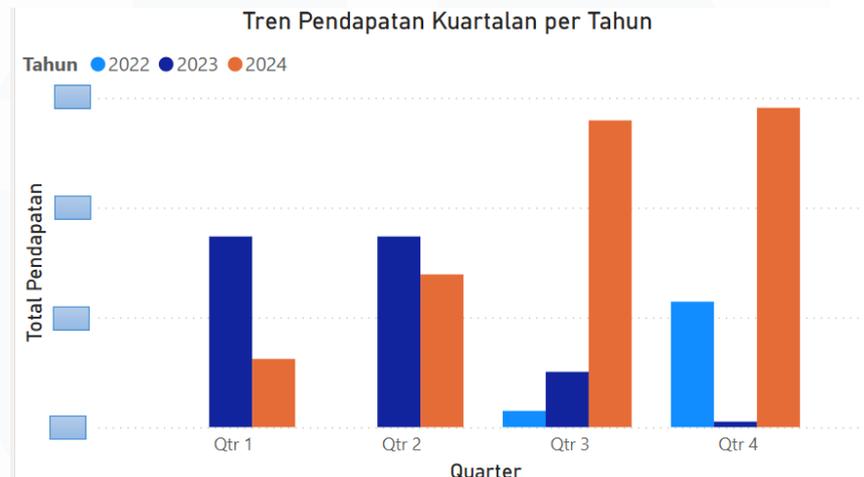
Gambar 3.61 Grafik Tren Pendapatan, Laba Kotor dan Laba Bersih per Bulan

Gambar 3.61 di atas menyajikan visualisasi tren bulanan yang membandingkan tiga metrik keuangan, yaitu: Total Pendapatan, Laba Kotor, dan Laba Bersih. Grafik ini memberikan gambaran yang jelas mengenai alur profitabilitas perusahaan dari pendapatan hingga laba akhir.

Dapat diamati bahwa kurva Total Pendapatan (biru muda) dan Laba Kotor (biru tua) bergerak secara paralel, di mana keduanya secara konsisten berada di wilayah positif. Hal ini menunjukkan bahwa aktivitas penjualan inti perusahaan mampu menghasilkan keuntungan kotor yang sehat setelah dikurangi Harga Pokok Penjualan (HPP).

Namun, wawasan yang paling krusial datang dari kurva Laba Bersih (oranye). Garis ini menunjukkan nilai yang secara konsisten berada jauh di wilayah negatif sepanjang tahun, yang menandakan perusahaan beroperasi dalam kondisi defisit yang signifikan setiap bulan. Kesenjangan yang sangat besar antara Laba Kotor yang positif dan Laba Bersih yang negatif ini menjadi bukti visual yang kuat bahwa beban biaya operasional sangat tinggi,

menghabiskan seluruh laba kotor yang dihasilkan dan mendorong perusahaan ke dalam kerugian.

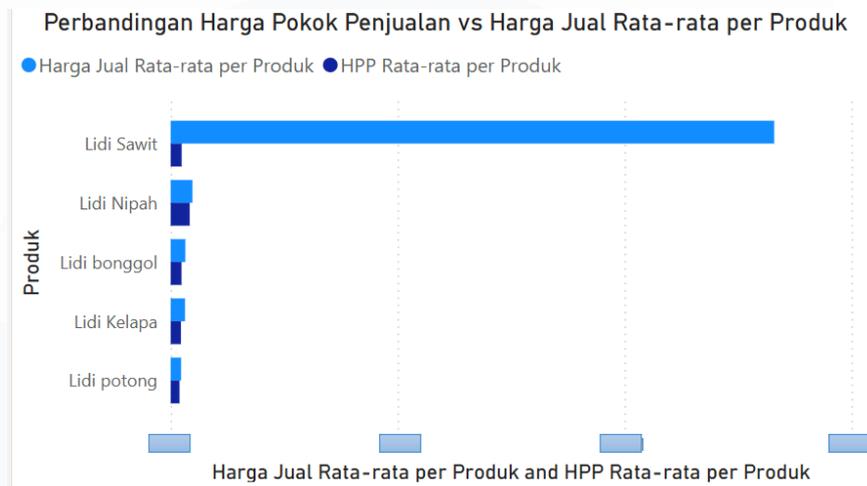


Gambar 3.62 Grafik Tren Pendapatan Kuartalan per Tahun

Gambar 3.62 di atas menyajikan perbandingan tren pendapatan per kuartal untuk periode tahun 2022 hingga 2024, yang mengungkap adanya perubahan pola pendapatan secara signifikan.

Visualisasi ini menunjukkan bahwa pada tahun 2022, pendapatan perusahaan terkonsentrasi pada paruh pertama tahun (Q1 dan Q2), sebelum mengalami penurunan drastis di paruh kedua. Sebaliknya, pola ini berbalik mulai tahun 2023, di mana pendapatan justru menunjukkan akselerasi signifikan pada paruh kedua, khususnya pada Q3 dan Q4. Tren penguatan pada akhir tahun ini berlanjut dan mengalami eskalasi yang luar biasa pada tahun 2024, di mana pendapatan pada Q3 dan Q4 tercatat sebagai yang tertinggi selama periode tiga tahun.

Secara keseluruhan, grafik ini tidak hanya mengonfirmasi adanya pertumbuhan pendapatan dari tahun ke tahun, tetapi juga menyoroti adanya pergeseran strategis atau musiman dalam model bisnis, yang kini semakin berfokus pada pencapaian target di semester kedua.

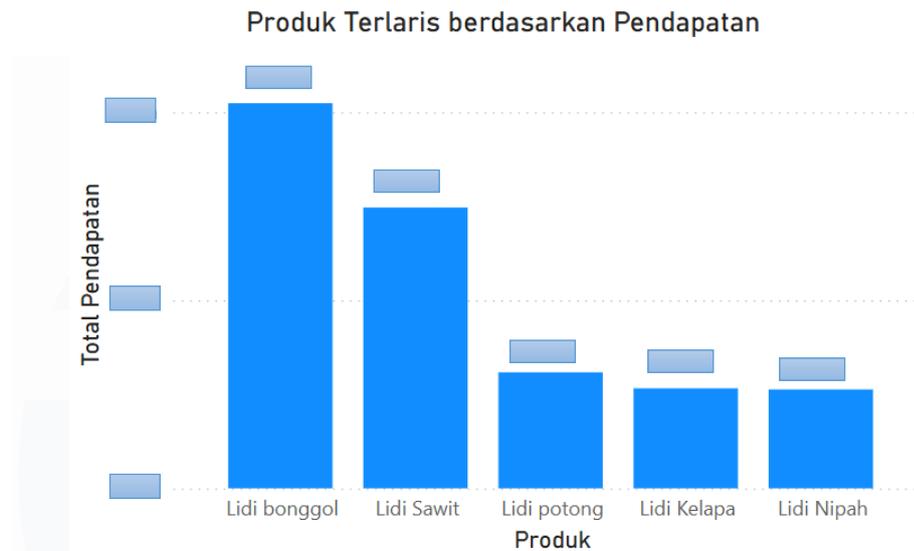


Gambar 3.63 Grafik Perbandingan HPP vs Harga Jual Rata-rata

Gambar 3.63 di atas menyajikan perbandingan visual antara Harga Pokok Penjualan (HPP) rata-rata dengan Harga Jual rata-rata untuk setiap produk utama perusahaan. Grafik ini secara jelas mengilustrasikan profitabilitas kotor dari masing-masing produk.

Dari visualisasi tersebut, dapat disimpulkan bahwa semua produk yang dijual memiliki harga jual yang lebih tinggi dari HPP-nya, yang menandakan bahwa seluruh lini produk memberikan kontribusi laba kotor bagi perusahaan. Namun, wawasan yang paling menonjol adalah peran strategis yang berbeda antar produk.

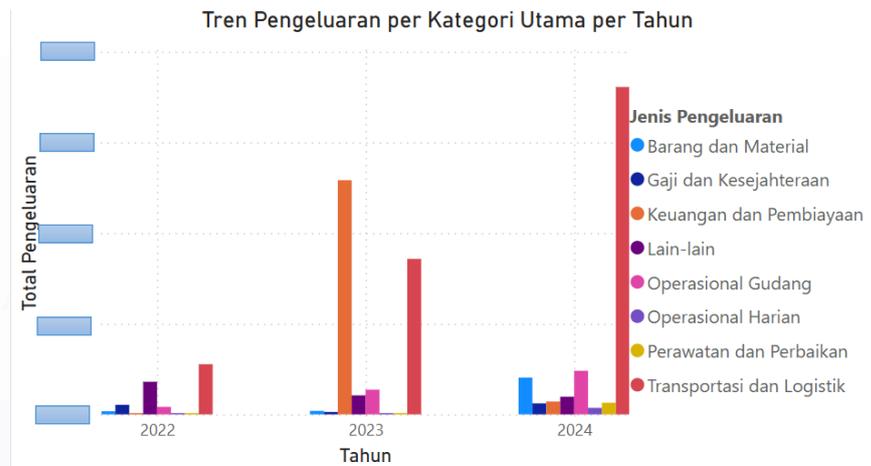
Lidi Sawit secara definitif menonjol dengan harga jual rata-rata yang jauh melampaui produk lainnya, serta memiliki selisih keuntungan per unit yang paling signifikan. Hal ini menegaskan perannya sebagai penggerak margin utama, yang berkontribusi pada kesehatan finansial perusahaan. Sementara itu, produk lain menunjukkan selisih keuntungan per unit yang lebih kecil, yang mengindikasikan kemungkinan peran mereka sebagai penggerak volume penjualan.



Gambar 3.64 Grafik Produk Terlaris berdasarkan Pendapatan

Gambar 3.64 di atas menampilkan peringkat produk berdasarkan total pendapatan yang dihasilkan selama periode 2022-2024. Visualisasi ini secara gamblang menunjukkan adanya fenomena konsentrasi produk yang sangat signifikan. Dari grafik tersebut, dapat disimpulkan:

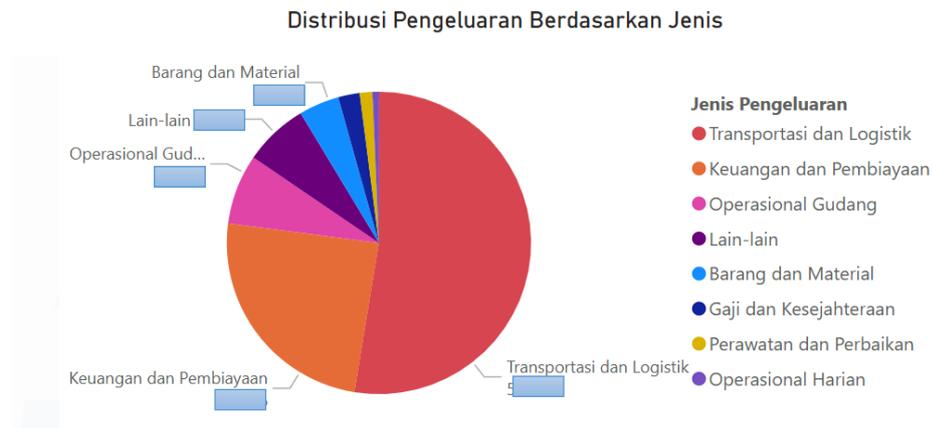
1. Lidi bonggol secara definitif berfungsi sebagai tulang punggung pendapatan utama perusahaan.
2. Lidi Sawit menempati posisi kedua sebagai kontributor pendapatan terbesar.
3. Terdapat kesenjangan yang sangat besar antara kontribusi pendapatan dari dua produk teratas ini dibandingkan dengan produk-produk lainnya.
4. Hal ini menegaskan bahwa mayoritas absolut dari pendapatan perusahaan ditopang oleh segelintir produk unggulan, sebuah kondisi ketergantungan karena dapat memunculkan risiko strategis.



Gambar 3.65 Grafik Tren Pengeluaran per Kategori Utama per Tahun

Gambar 3.65 di atas menyajikan visualisasi tren pengeluaran tahunan yang dipecah berdasarkan kategori utamanya dari tahun 2022 hingga 2024. Grafik ini secara jelas mengilustrasikan komposisi dan evolusi struktur biaya perusahaan. Dari visualisasi tersebut, dapat ditarik beberapa kesimpulan utama, yaitu:

- Grafik ini menunjukkan bahwa struktur biaya operasional perusahaan sangat terkonsentrasi. Sebagian besar pengeluaran secara dominan bersumber hanya dari dua kategori utama, yaitu Transportasi dan Logistik dan Keuangan dan Pembiayaan.
- Terjadi lonjakan total pengeluaran yang sangat signifikan pada tahun 2023. Eskalasi biaya yang luar biasa ini hampir seluruhnya didorong oleh peningkatan masif pada kedua kategori dominan tersebut, sementara tahun 2022 dan 2024 menunjukkan tingkat pengeluaran yang relatif lebih rendah dan terkendali.
- Lonjakan biaya pada tahun 2023 ini menyoroti adanya hubungan langsung antara akselerasi pertumbuhan pendapatan yang juga terjadi pada tahun yang sama dengan pembengkakan biaya logistik dan finansial yang diperlukan untuk mendukung ekspansi tersebut.

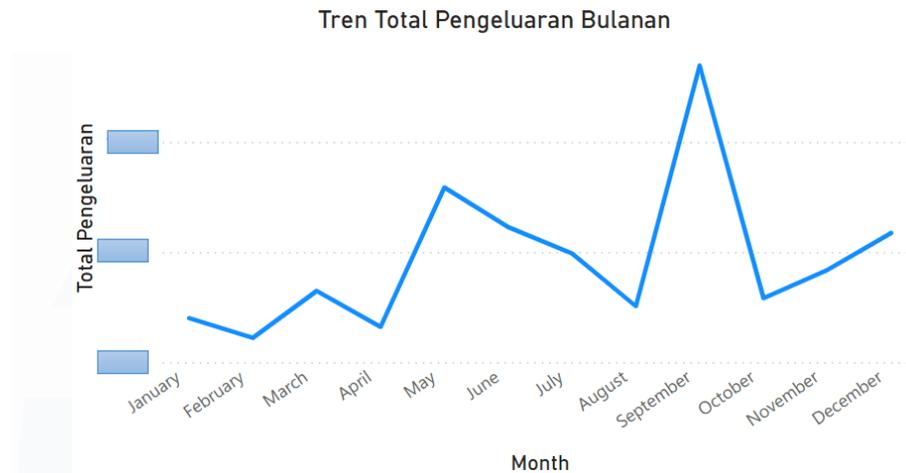


Gambar 3.66 Grafik Distribusi Pengeluaran Berdasarkan Jenis

Gambar 3.66 di atas menyajikan visualisasi distribusi pengeluaran berdasarkan jenis dalam bentuk diagram pai, yang memberikan gambaran proporsional dari setiap kategori biaya terhadap total pengeluaran selama periode 2022-2024. Grafik ini secara gamblang mengonfirmasi temuan mengenai struktur biaya yang sangat terkonsentrasi.

1. Transportasi dan Logistik merupakan komponen pengeluaran terbesar yang menyumbang lebih dari separuh total biaya.
2. Kategori terbesar kedua adalah Keuangan dan Pembiayaan.
3. Secara kumulatif, kedua kategori ini mendominasi lebih dari tiga perempat total pengeluaran perusahaan. Kategori lainnya seperti Operasional Gudang, Lain-lain, dan Barang dan Material memiliki porsi yang jauh lebih kecil.

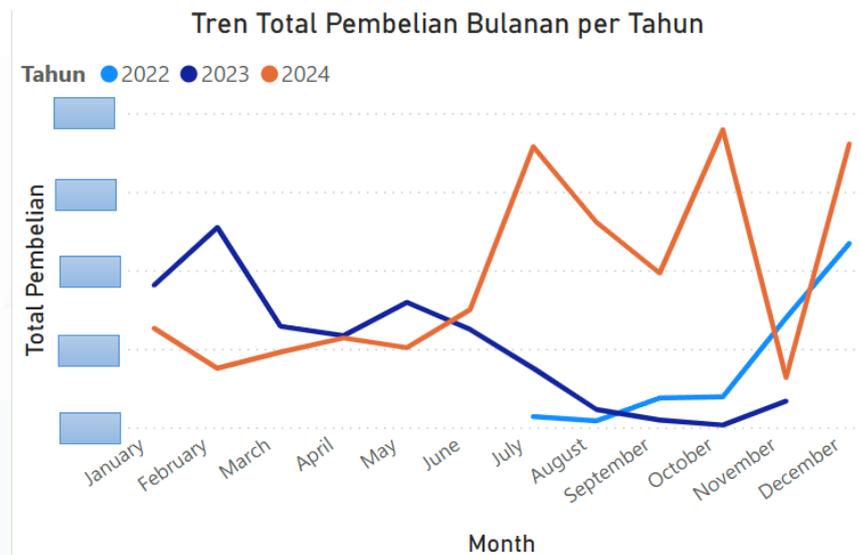
Visualisasi ini menegaskan bahwa setiap upaya efisiensi biaya harus difokuskan secara strategis pada area transportasi, logistik, dan pembiayaan untuk dapat memberikan dampak yang signifikan terhadap profitabilitas perusahaan.



Gambar 3.67 Grafik Tren Total Pengeluaran Bulanan

Gambar 3.67 di atas menampilkan visualisasi tren total pengeluaran bulanan, yang merupakan agregasi dari seluruh biaya operasional per bulan selama periode 2022-2024. Grafik ini bertujuan untuk mengidentifikasi pola musiman atau anomali temporal dalam struktur biaya perusahaan. Dari visualisasi tersebut, dapat ditarik beberapa kesimpulan:

1. Tren pengeluaran menunjukkan adanya volatilitas yang signifikan sepanjang tahun.
2. Wawasan yang paling menonjol adalah adanya satu anomali temporal yang sangat signifikan, yaitu lonjakan biaya yang luar biasa tajam yang berpuncak pada bulan September.
3. Puncak pengeluaran pada bulan September ini secara logis berkorelasi dengan eskalasi biaya yang teridentifikasi terjadi pada tahun 2023. Hal ini mengindikasikan adanya sebuah peristiwa atau proyek dengan intensitas modal tinggi yang terkonsentrasi pada periode tersebut, kemungkinan besar untuk mendukung ekspansi pendapatan yang juga terjadi pada paruh kedua tahun itu.



Gambar 3.68 Grafik Tren Total Pembelian Bulanan per Tahun

Gambar 3.68 di atas menyajikan perbandingan tren total pembelian bahan baku per bulan, yang dipecah untuk setiap tahun dari 2022 hingga 2024. Visualisasi ini secara jelas menunjukkan adanya evolusi dan volatilitas yang signifikan dalam pola pengadaan barang perusahaan. Dari grafik tersebut, dapat diamati:

1. Pada tahun 2022 (garis biru tua), aktivitas pembelian terkonsentrasi pada paruh pertama tahun, yang kemudian menurun drastis di semester kedua.
2. Pola ini berubah total pada tahun-tahun berikutnya. Tahun 2023 (garis biru muda) menunjukkan adanya eskalasi pembelian yang bersifat reaktif pada kuartal terakhir untuk mendukung lonjakan permintaan yang terjadi pada periode tersebut.
3. Sementara itu, tahun 2024 (garis oranye) menampilkan pola yang paling tidak stabil dengan puncak-puncak pembelian yang sangat tinggi terjadi di paruh kedua tahun, terutama pada bulan Juli dan Oktober.

Secara keseluruhan, grafik ini mengilustrasikan pergeseran strategi pengadaan dari yang sebelumnya terjadwal di awal tahun menjadi lebih reaktif

dan terkonsentrasi di akhir tahun, sejalan dengan pergeseran pola pendapatan perusahaan.

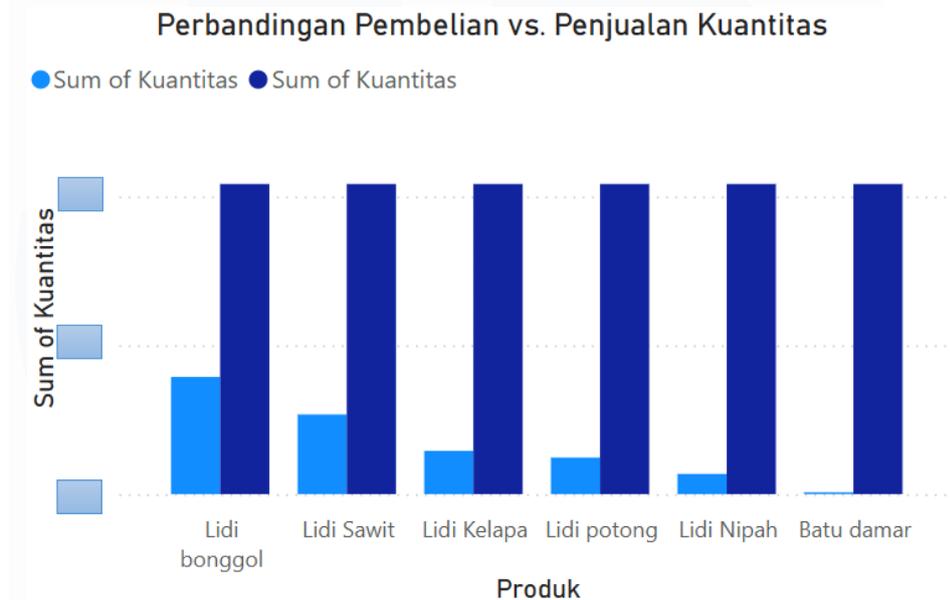


Gambar 3.69 Grafik Produk Baku yang Paling Banyak Dibeli

Gambar 3.69 di atas menampilkan peringkat produk baku berdasarkan total nilai pembelian selama periode 2022-2024, yang mengilustrasikan di mana sebagian besar modal pengadaan perusahaan dialokasikan. Dari visualisasi tersebut, dapat disimpulkan:

1. Grafik ini secara jelas menunjukkan bahwa struktur biaya pengadaan bahan baku perusahaan memiliki tingkat konsentrasi yang tinggi.
2. Produk Lidi bonggol dan Lidi Sawit merupakan dua komponen terbesar dalam total belanja bahan baku perusahaan.
3. Pola pengadaan ini secara langsung merefleksikan struktur pendapatan perusahaan. Produk yang paling banyak dibeli adalah bahan baku untuk produk yang paling banyak dijual, yang menegaskan bahwa strategi pengadaan sangat dipengaruhi oleh permintaan pasar dan fokus penjualan perusahaan.

- Produk-produk lain seperti Lidi Kelapa, Lidi Nipah, dan Lidi potong memiliki porsi pembelian yang jauh lebih kecil, sejalan dengan peran mereka yang lebih sekunder dalam portofolio penjualan.



Gambar 3.70 Grafik Perbandingan Pembelian vs Penjualan Kuantitas

Gambar 3.70 di atas menyajikan perbandingan langsung antara total kuantitas bahan baku yang dibeli (biru tua) dengan total kuantitas produk jadi yang terjual (biru muda) untuk setiap lini produk. Visualisasi ini mengungkap pengetahuan terkait efisiensi proses produksi dan manajemen inventaris. Dari grafik tersebut, dapat disimpulkan:

- Terdapat selisih yang sangat signifikan, di mana untuk semua lini produk utama, jumlah kuantitas bahan baku yang dibeli jauh melampaui jumlah kuantitas produk jadi yang berhasil terjual.
- Fenomena ini secara kuat mengindikasikan adanya dua kemungkinan utama yang perlu diinvestigasi lebih lanjut oleh manajemen. Pertama adanya tingkat penyusutan material yang tinggi selama proses produksi dari bahan mentah menjadi produk jadi. Kedua adanya kebijakan strategis untuk melakukan penumpukan stok bahan baku sebagai

persediaan untuk mengantisipasi volatilitas permintaan atau pasokan di masa depan.

3. Dengan demikian, grafik ini menjadi alat analisis yang vital untuk mendorong evaluasi terhadap efisiensi produksi dan strategi manajemen inventaris perusahaan.

3.2.6 Evaluasi dan Finalisasi Dashboard

Setelah seluruh metrik kinerja utama (KPI) berhasil dirumuskan, alur kerja memasuki tahap evaluasi dan finalisasi *dashboard*. Pada fase ini, berbagai komponen visual seperti grafik, diagram, dan kartu KPI yang telah dirancang kemudian dirakit dan disusun secara sistematis ke dalam beberapa halaman tematik untuk menyajikan pemaparan bisnis yang terstruktur dan mudah dipahami. *Dashboard* interaktif ini dirancang untuk memberdayakan manajemen dengan kemampuan untuk memfilter dan menelusuri data, guna mendukung pengambilan keputusan yang tangkas dan berbasis bukti.

Bagian berikut ini akan menjabarkan hasil evaluasi dari setiap halaman *dashboard* yang telah dikembangkan, merangkum wawasan-wawasan strategis utama yang terungkap dari analisis visual data keuangan, penjualan, pengeluaran, dan pembelian PT Djaya Universal Indonesia selama periode 2022-2024.

3.2.6.1 Finalisasi Dashboard Tren Pendapatan, Laba Kotor dan Laba Bersih per Bulan



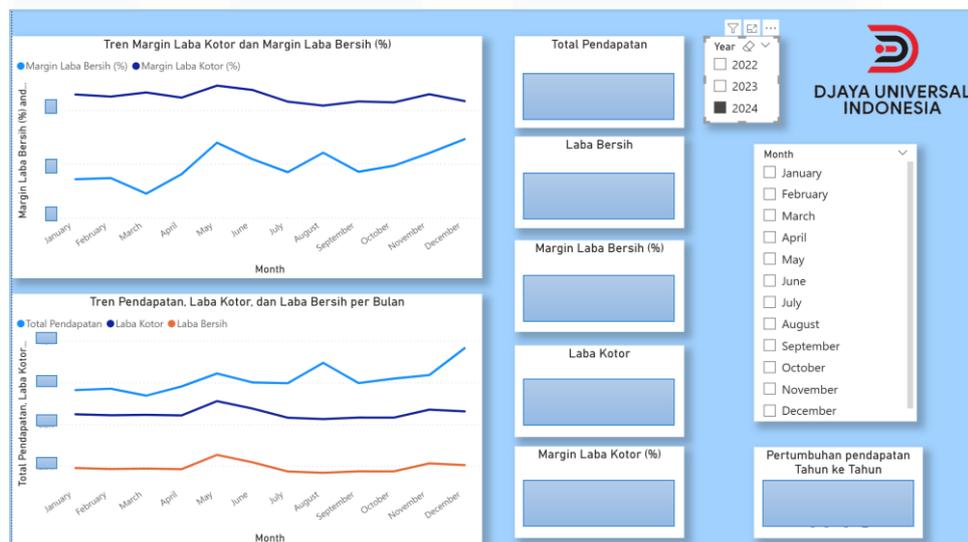
Gambar 3.71 Dashboard Tren Pendapatan, Laba Kotor, dan Laba Bersih

Gambar 3.71 diatas menunjukkan analisis *dashboard* visual yang merangkum data keuangan perusahaan dari tahun 2022 hingga 2024, dapat disimpulkan bahwa PT Djaya Universal Indonesia menunjukkan sebuah performa finansial yang dualistis. Pada tingkat operasional inti, perusahaan menampilkan landasan yang solid, yang dibuktikan oleh kapabilitasnya dalam menghasilkan pendapatan kotor dan mempertahankan margin laba kotor pada level yang sehat.

Kinerja pendapatan ini secara konsisten menegaskan bahwa strategi penetapan harga dan manajemen harga pokok penjualan telah dieksekusi secara efektif. Pemahaman terhadap kekuatan pada level laba kotor ini menjadi fondasi penting, karena menunjukkan bahwa model bisnis perusahaan secara inheren mampu menciptakan nilai yang signifikan dari aktivitas penjualan utamanya.

Meskipun demikian, analisis yang lebih terperinci pada tingkat profitabilitas akhir mengungkap sebuah wawasan krusial terkait tantangan besar dalam efisiensi operasional. Ditemukan bahwa laba kotor yang solid

tersebut hampir seluruhnya terkikis oleh beban biaya operasional, yang mengakibatkan margin laba bersih jatuh ke tingkat yang sangat kritis dan mendekati titik impas. Hal ini secara gamblang mengindikasikan adanya struktur biaya yang tidak efisien, yang menyebabkan perusahaan secara frekuen beroperasi dalam kondisi defisit pada sebagian besar periode bulanan. Kinerja margin laba kotor yang sehat dapat dijadikan sebagai tolok ukur internal yang kontras, menyoroti besarnya potensi profitabilitas yang hilang akibat beban operasional.



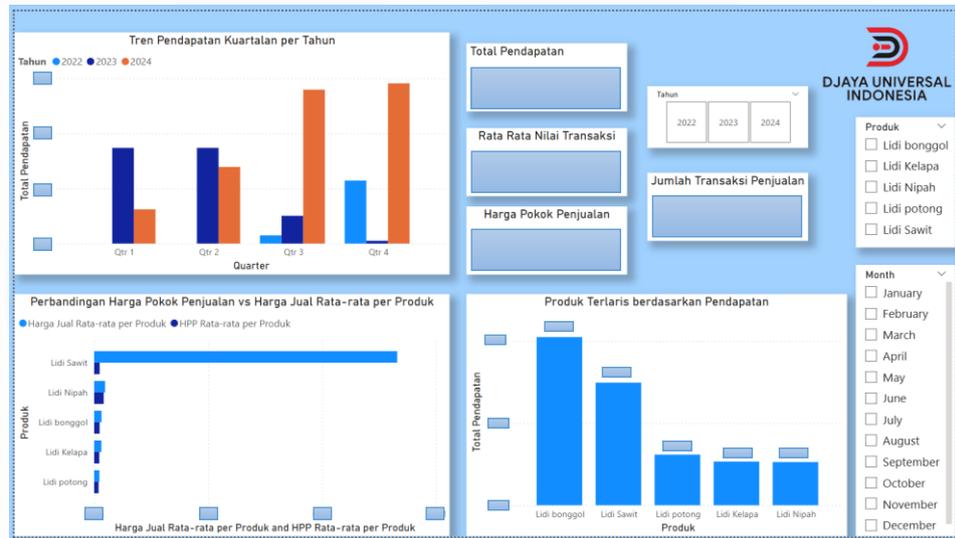
Gambar 3.72 Contoh Dashboard 2024 Tren Pendapatan, Laba Kotor, dan Laba Bersih

Gambar 3.72 diatas menunjukkan analisis *dashboard* yang berfokus pada kinerja keuangan PT Djaya Universal Indonesia tahun 2024, teridentifikasi adanya sebuah tren pertumbuhan pendapatan yang dinamis secara bulanan. Kinerja pendapatan laba bersih menunjukkan adanya beberapa puncak aktivitas yang signifikan, terutama pada bulan Mei dan September, sebelum akhirnya mencapai akhir tahun di bulan Desember. Tren progresif sepanjang tahun ini menandakan keberhasilan strategi perusahaan dalam meningkatkan volume penjualan.

Oleh karena itu, menjadi sebuah langkah yang imperatif bagi manajemen untuk memprioritaskan program efisiensi biaya secara radikal

dan komprehensif, guna memulihkan kesehatan finansial dan memastikan keberlanjutan perusahaan dalam jangka panjang.

3.2.6.2 Finalisasi Dashboard Analisis Pendapatan & Penjualan

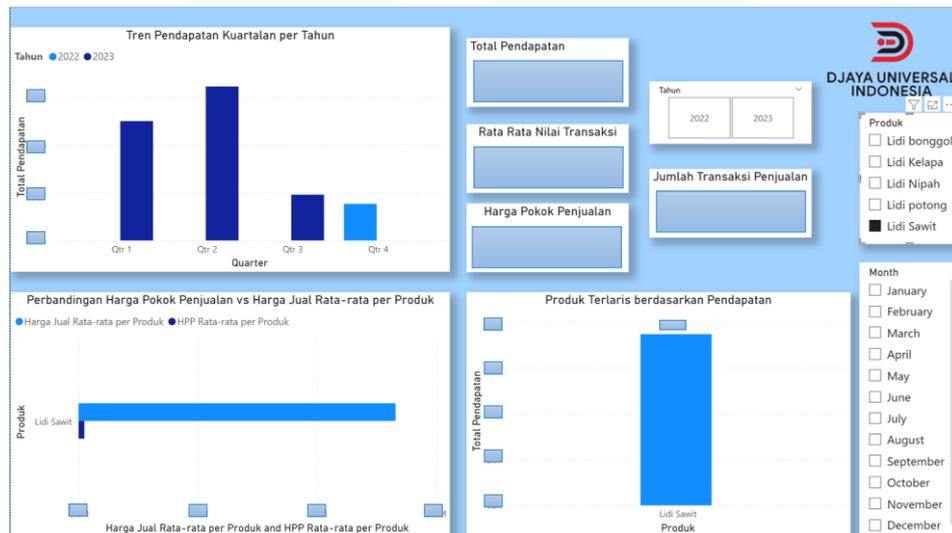


Gambar 3.73 Dashboard Analisis Pendapatan & Penjualan

Gambar 3.73 diatas menunjukkan analisis terperinci terhadap *dashboard* kinerja keuangan PT Djaya Universal Indonesia periode 2022-2024, terungkap sebuah model bisnis dengan landasan finansial yang kokoh. Kekuatan utama perusahaan bersumber dari model penjualan yang berfokus pada transaksi bernilai tinggi, yang mengindikasikan adanya hubungan kemitraan strategis dengan basis klien yang loyal dan terseleksi. Struktur pendapatan perusahaan menunjukkan fenomena konsentrasi produk yang sangat signifikan, di mana mayoritas absolut dari total pendapatan ditopang oleh segelintir produk unggulan. Lidi bonggol berfungsi sebagai tulang punggung pendapatan, sementara Lidi Sawit memegang peranan sebagai kontributor terbesar kedua.

Ketergantungan ini, meskipun menciptakan efisiensi dalam produksi dan pemasaran, yang memunculkan risiko strategis jika terjadi perubahan besar pada pasar atau rantai pasok produk-produk kunci tersebut. Oleh karena itu, pemahaman menyeluruh terhadap struktur ini bukan hanya

menjadi aset, melainkan titik awal yang krusial bagi seluruh strategis, termasuk mitigasi risiko, variasi produk di masa depan, dan alokasi modal untuk riset dan pengembangan.

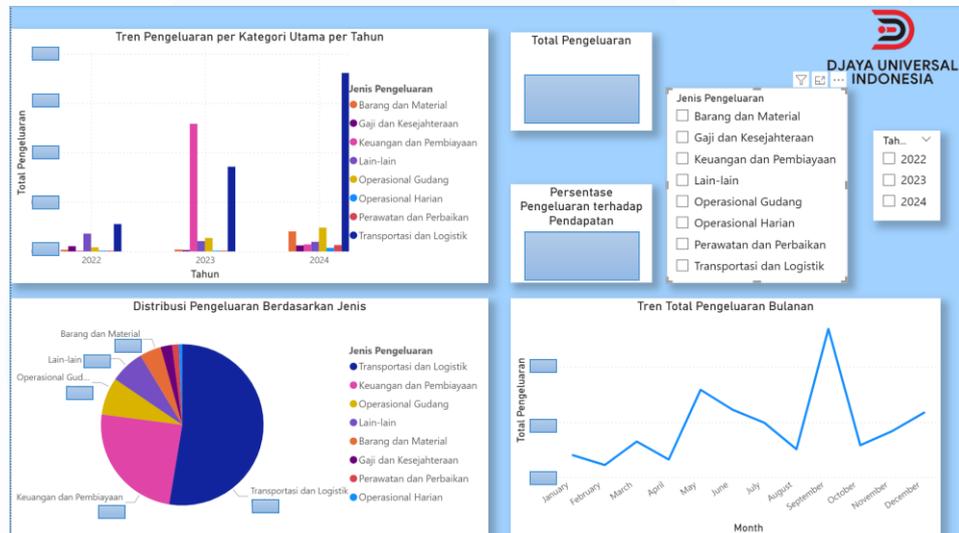


Gambar 3.74 Contoh Dashboard Analisis Pendapatan & Penjualan Produk Lidi Sawit

Gambar 3.74 diatas menunjukkan analisis *dashboard* yang difokuskan pada kinerja produk Lidi Sawit selama periode 2022-2023, Data secara gamblang menunjukkan bahwa Lidi Sawit dijual bernilai tinggi, yang menegaskan posisinya sebagai produk unggulan perusahaan. Analisis perbandingan antara harga jual rata-rata dengan harga pokok penjualan secara visual mengonfirmasi adanya margin laba kotor yang sangat baik, yang menjadi dorong utama profitabilitas perusahaan. Namun, tren pendapatan kuartalan untuk produk ini menunjukkan pola yang tidak menentu dan tidak konsisten antar tahun, kinerja yang kuat pada kuartal kedua tahun 2022 sangat kontras dengan lonjakan pendapatan masif yang terkonsentrasi pada kuartal keempat tahun 2023.

Meskipun demikian, analisis pada tingkat profitabilitas per unit mengungkap adanya sebuah sinergi strategis yang kompleks dan menjadi inti dari keunggulan kompetitif perusahaan.

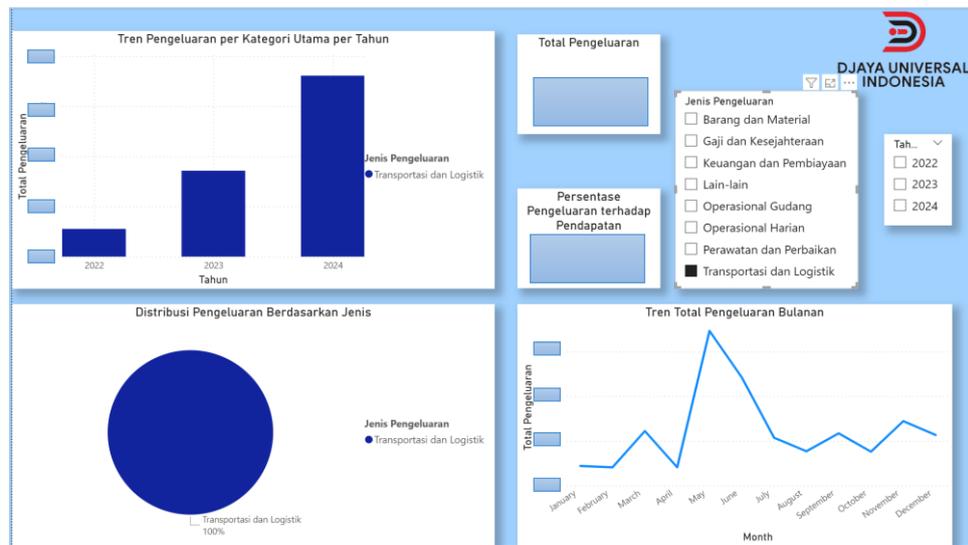
3.2.6.3 Finalisasi Dashboard Analisis Pengeluaran & Biaya



Gambar 3.75 Dashboard Analisis Pengeluaran & Biaya

Gambar 3.75 diatas menunjukkan analisis terhadap *dashboard* kinerja pengeluaran PT Djaya Universal Indonesia periode 2022-2024, teridentifikasi sebuah struktur biaya operasional yang sangat terkonsentrasi. Data secara gamblang menunjukkan bahwa lebih dari tiga perempat dari total pengeluaran perusahaan secara dominan bersumber hanya dari dua kategori utama, yaitu Transportasi dan Logistik, yang menyumbang lebih dari separuh total biaya, serta Keuangan dan Pembiayaan. Ketergantungan yang signifikan pada dua pengeluaran ini menandakan bahwa efisiensi pada kedua area tersebut memegang peranan krusial bagi kesehatan profitabilitas perusahaan.

Analisis lebih lanjut pada tren tahunan mengungkap bahwa struktur biaya yang timpang ini tidak bersifat konstan, melainkan sangat dipengaruhi oleh pengeluaran yang terjadi secara spesifik pada tahun 2023, di mana terjadi kenaikan biaya yang luar biasa pada kedua kategori dominan tersebut, sementara tahun lainnya menunjukkan tingkat pengeluaran yang relatif stabil dan terkendali.

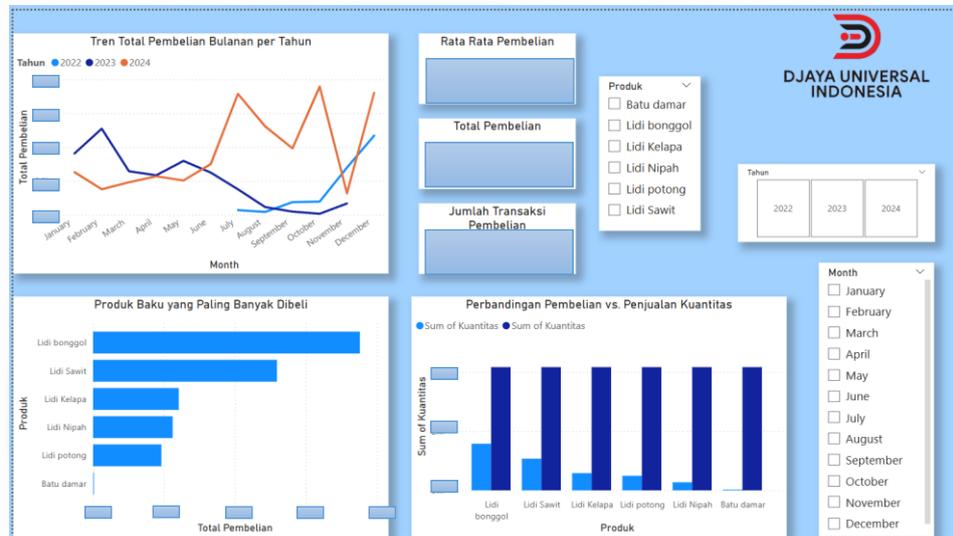


Gambar 3.76 contoh Dashboard Analisis Pengeluaran & Biaya Transportasi dan Logistik

Gambar 3.76 diatas menunjukkan analisis *dashboard* yang difokuskan pada kategori pengeluaran Transportasi dan Logistik dari tahun 2022 hingga 2024, teridentifikasi bahwa biaya memegang peran sangat krusial dalam struktur finansial perusahaan. Data secara gamblang menunjukkan bahwa pengeluaran ini sangat signifikan dari total pendapatan perusahaan, yang menegaskan bahwa ini merupakan faktor utama profitabilitas secara keseluruhan.

Meskipun demikian, analisis pada tren bulanan mengungkap bahwa pengeluaran tahunan tersebut berpuncak pada satu pengeluaran yang sangat signifikan, yaitu lonjakan biaya yang terjadi pada bulan September. Puncak pengeluaran ini, yang secara logis berkorelasi dengan biaya di tahun 2023, mengindikasikan adanya sebuah proyek dengan intensitas modal tinggi yang terkonsentrasi pada periode tersebut, kemungkinan besar untuk mendukung ekspansi pendapatan yang juga terjadi pada paruh kedua tahun itu.

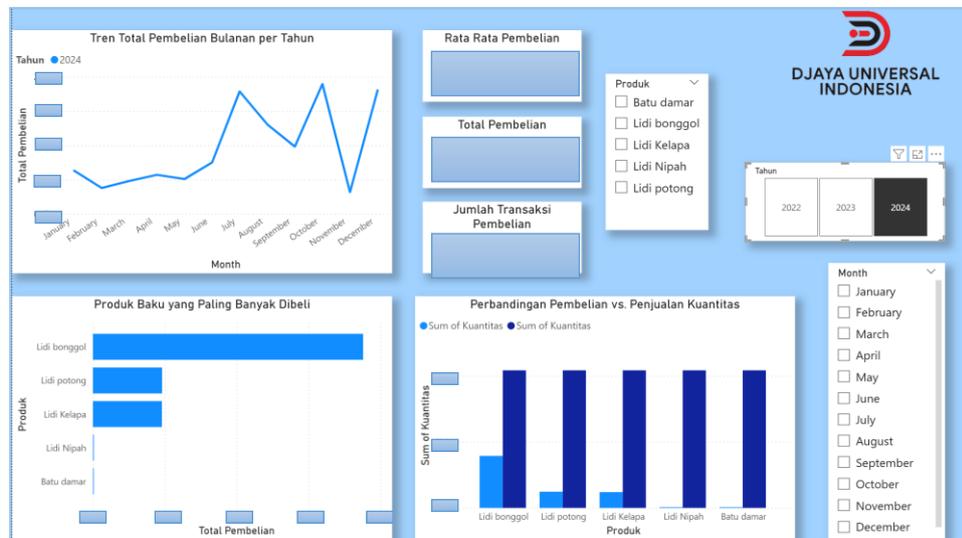
3.2.6.4 Finalisasi Dashboard Analisis Pembelian



Gambar 3.77 Dashboard Analisis Pembelian

Gambar 3.77 diatas menunjukkan analisis terperinci terhadap *dashboard* kinerja pengadaan PT Djaya Universal Indonesia periode 2022-2024, teridentifikasi sebuah model operasional dan struktur pengadaan bahan baku yang sangat khas. Data secara gamblang menunjukkan adanya pola pengadaan yang bersifat frekuen dengan nilai per transaksi yang lebih kecil, yang berkontras tajam dengan pola penjualan yang bersifat jarang namun bernilai masif.

Hal ini mengindikasikan sebuah alur bisnis yang berpusat pada proses pengolahan, di mana bahan baku dibeli secara berkala untuk diolah menjadi produk jadi bernilai tinggi. Struktur biaya pengadaan juga menunjukkan tingkat konsentrasi yang tinggi, yang secara langsung merefleksikan struktur pendapatan perusahaan, produk Lidi bonggol dan Lidi Sawit tidak hanya mendominasi penjualan tetapi juga menjadi komponen terbesar dalam total belanja bahan baku.



Gambar 3.78 Contoh Dashboard Analisis Pembelian Tahun 2024

Gambar 3.78 diatas menunjukkan analisis *dashboard* kinerja pengadaan yang difokuskan pada tahun 2024, teridentifikasi adanya sebuah aktivitas pembelian yang sangat signifikan, di mana total nilai pembelian pada tahun ini merepresentasikan mayoritas dari total pembelian. Kinerja ini menunjukkan bahwa tahun 2024 merupakan periode ekspansi atau pemenuhan permintaan berskala besar. Struktur pembelian pada tahun ini juga menunjukkan kenaikan yang jauh lebih dominan, dengan alokasi belanja yang sepenuhnya didominasi oleh bahan baku Lidi bonggol.

Meskipun demikian, analisis perbandingan kuantitas antara material yang dibeli dengan produk yang dijual mengungkap sebuah wawasan krusial terkait efisiensi proses produksi dan manajemen inventaris. Ditemukan adanya selisih yang sangat signifikan, di mana jumlah kuantitas bahan baku yang dibeli jauh melampaui jumlah kuantitas produk jadi yang terjual untuk semua lini produk. Fenomena ini secara kuat mengindikasikan dua kemungkinan utama: pertama, adanya tingkat penyusutan material selama proses produksi dan kedua, adanya kebijakan penumpukan stok bahan baku sebagai persediaan strategis untuk mengantisipasi volatilitas permintaan atau pasokan.

3.2.7 Serah Terima Kerja

Menjelang berakhirnya periode magang pada pertengahan bulan Juni, dilaksanakan sebuah proses serah terima kerja yang formal dan terstruktur sebagai tahap akhir dari keseluruhan proyek. Proses ini bukan sekadar penyerahan hasil, melainkan sebuah transfer pengetahuan yang bertujuan untuk memastikan keberlanjutan dan kemandirian PT Djaya Universal Indonesia dalam memanfaatkan aset analitik yang telah dibangun. Seluruh komponen proyek diserahkan kepada pembimbing lapangan, yang mencakup beberapa aset digital utama.

Pertama, diserahkan *file dashboard* interaktif Microsoft Power BI yang telah final, yang dirancang sebagai alat eksplorasi mandiri bagi manajemen untuk mendukung pengambilan keputusan strategis.

Kedua, diserahkan pula seluruh skrip *Python* yang telah dikembangkan untuk proses pembersihan, integrasi, dan analisis data. Skrip ini didokumentasikan dengan baik untuk memastikan proses pengolahan data dapat diulang dan diaudit dengan mudah di kemudian hari.

Aset ketiga yang diserahkan adalah *file data_bersih.xlsx*, yang merupakan output akhir dari skrip *Python* dan berfungsi sebagai sumber data tunggal yang bersih dan terstruktur untuk *dashboard* Power BI. Terakhir, laporan magang ini sendiri menjadi bagian dari serah terima, berfungsi sebagai dokumentasi teknis dan panduan pengguna yang komprehensif. Laporan ini merinci seluruh metodologi, dari proses eksplorasi data mentah hingga interpretasi setiap visualisasi pada *dashboard*. Dengan demikian, proses serah terima ini dirancang untuk memberdayakan perusahaan agar dapat secara mandiri mengoperasikan, memelihara, dan bahkan mengembangkan lebih lanjut kerangka kerja analitis yang telah ada, sehingga memastikan nilai jangka panjang dari proyek magang ini.

3.2.8 Update Data Harian

Di samping alur kerja utama yang berfokus pada pengembangan proyek *dashboard* analitik, terdapat pula sebuah tanggung jawab operasional rutin yang dilaksanakan secara paralel dan konsisten, yaitu melakukan pembaruan data harian. Tugas ini menjadi krusial mengingat karakteristik bisnis ekspor yang memiliki aliran data kompleks dan bersifat *real-time*. Pelaksanaan tugas ini dimulai sejak pertengahan bulan April, setelah fondasi data historis selesai dibersihkan dan diintegrasikan, dan terus berlanjut hingga akhir periode magang pada bulan Juni.

Proses pembaruan ini melibatkan pengintegrasian catatan transaksi terbaru yang terjadi setiap harinya mencakup data penjualan, pembelian bahan baku, dan pengeluaran operasional ke dalam data yang digunakan untuk analisis. Tanggung jawab ini memastikan bahwa *dashboard* dan laporan yang dihasilkan tidak hanya berfungsi sebagai alat analisis historis, tetapi juga sebagai instrumen pemantauan kinerja yang relevan dan akurat. Dengan data yang selalu diperbarui, manajemen dapat memperoleh gambaran terkini mengenai kondisi operasional perusahaan. Tugas ini memberikan pengalaman langsung dalam mempelajari alur kerja pengelolaan data dalam siklus bisnis yang nyata dan menjadi kunci dalam mendukung proses pengambilan keputusan harian yang lebih cepat dan tanggap.

3.3 Kendala yang Ditemukan

Selama pelaksanaan proyek analisis dan visualisasi data di PT Djaya Universal Indonesia, ditemukan beberapa kendala teknis yang menjadi tantangan utama dalam proses pengolahan data. Kendala-kendala ini secara garis besar dapat dikelompokkan menjadi dua kategori utama:

1. Kompleksitas dan Fragmentasi Data Sumber: Kendala pertama dan paling fundamental adalah struktur data mentah yang sangat kompleks. Data historis perusahaan dari tahun 2022 hingga 2024 tidak tersimpan dalam format tabel yang sederhana, melainkan dalam *file* kerja akuntansi yang

berisi berbagai macam *sheet* seperti Jurnal Kas, Jurnal Pengeluaran, dan Invoice. Akibatnya, informasi yang relevan untuk satu pilar analisis (misalnya, pengeluaran) tersebar di beberapa lokasi berbeda, sehingga proses untuk mengidentifikasi, mengekstraksi, dan mengonsolidasikan data yang dibutuhkan menjadi sangat menantang dan memakan waktu.

2. Kualitas dan Integritas Data: Setelah data berhasil diekstraksi, kendala berikutnya yang dihadapi adalah terkait kualitas data itu sendiri. Ditemukan beberapa isu integritas data yang signifikan, antara lain:

- a. Nilai Kosong (*Missing Values*): Pada ketiga set data, ditemukan adanya baris-baris yang tidak lengkap atau mengandung nilai kosong. Keberadaan data yang tidak lengkap ini berisiko mengganggu akurasi perhitungan agregat.
- b. Data Duplikat (*Duplicate Entries*): Pemeriksaan data menunjukkan adanya entri data yang berulang atau duplikat secara signifikan, terutama pada data pengeluaran dan data pembelian. Data duplikat ini dapat menyebabkan hasil analisis menjadi bias, seperti perhitungan total biaya yang lebih tinggi dari seharusnya.
- c. Kolom Tidak Relevan: Data gabungan awal juga mengandung beberapa kolom "Unnamed" yang tidak memiliki informasi dan hanya berisi nilai kosong, sehingga mengotori struktur data dan perlu untuk dibersihkan.

3.4 Solusi atas Kendala yang Ditemukan

Untuk mengatasi setiap kendala yang telah diidentifikasi, diterapkan serangkaian solusi teknis yang sistematis dan terprogram menggunakan bahasa pemrograman *Python* dan *library Pandas*. Solusi-solusi tersebut adalah sebagai berikut:

1. Restrukturisasi Data dan Otomatisasi Impor: Untuk mengatasi masalah kompleksitas dan fragmentasi data, solusi dua tahap diimplementasikan.

Tahap pertama adalah melakukan restrukturisasi data secara manual, di mana data yang relevan dari berbagai *sheet* akuntansi diekstraksi dan disusun ulang ke dalam sebuah *file* Excel baru yang lebih sederhana dengan tiga *sheet* terstandarisasi: 'Pengeluaran', 'Penjualan', dan 'Pembelian'. Tahap kedua, untuk meningkatkan efisiensi proses impor data tahunan yang sudah terstruktur ini, sebuah fungsi *Python* kustom bernama `read_selected_sheets` dikembangkan untuk membaca dan memberi label pada data dari semua *file* secara otomatis.

2. Implementasi Prosedur Pembersihan Data: Untuk menangani masalah kualitas dan integritas data, serangkaian metode pembersihan data dari *library Pandas* diterapkan secara berurutan:
 - 1) Penghapusan Nilai Kosong: Seluruh baris yang mengandung nilai kosong pada ketiga set data dibersihkan dengan menggunakan fungsi `.dropna()`. Langkah ini memastikan bahwa setiap baris data yang akan dianalisis telah lengkap dan valid.
 - 2) Penghapusan Data Duplikat: Untuk mengatasi entri ganda, fungsi `.drop_duplicates()` diterapkan pada data pengeluaran dan pembelian. Solusi ini secara efektif menghapus semua catatan yang berulang, sehingga memastikan integritas data dan akurasi hasil analisis akhir.
 - 3) Penyaringan Kolom: Kolom-kolom "Unnamed" yang tidak relevan dan tidak berisi informasi penting berhasil dihilangkan dari set data dengan menggunakan metode penyaringan kolom, sehingga menghasilkan struktur data yang bersih dan terfokus pada informasi yang dibutuhkan.