

## BAB III

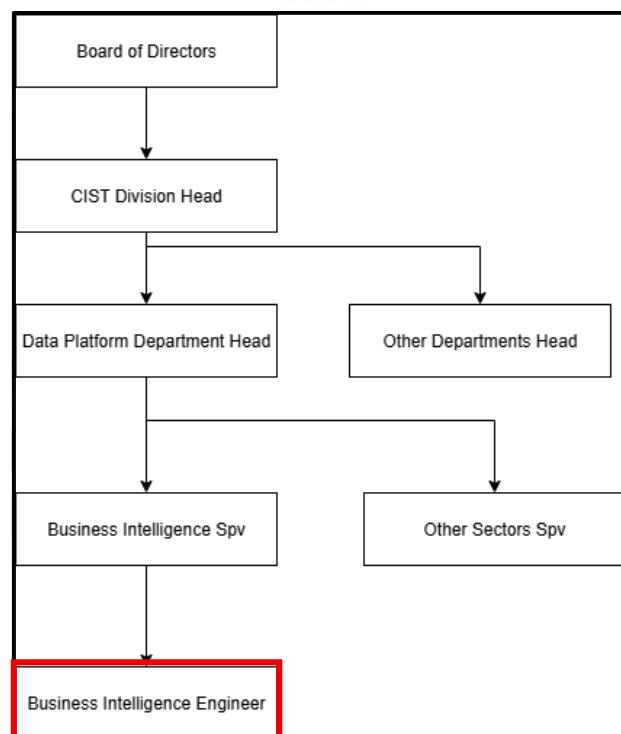
### PELAKSANAAN KERJA

#### 3.1 Kedudukan dan Koordinasi

Bagian ini berisi informasi mengenai posisi dan alur koordinasi dengan pembimbing lapangan pada saat pelaksanaan suatu proyek yang diberikan saat kegiatan magang sedang berlangsung.

##### 3.1.1 Kedudukan

Berdasarkan Gambar 3.1, kedudukan kegiatan magang berada pada bagian dengan persegi panjang berwarna merah. Sebagai seorang Business Intelligence Engineer Intern, setiap hasil kegiatan selalu perlu dilaporkan kepada Business Intelligence Supervisor untuk di review. Jika hasil kegiatan sudah di approve, barulah hasil tersebut dilaporkan ke pihak kepala departemen.



Gambar 3.1 Kedudukan Kegiatan Magang

Sumber: [www.sera.astra.co.id](http://www.sera.astra.co.id) [11]

Berikut adalah penjelasan mengenai hirarki pada diagram tersebut:

1. *Board of Directors* (Dewan Direksi)

Dewan Direksi merupakan tingkatan tertinggi di perusahaan ini. Mereka bertanggung jawab atas segala keputusan strategis dan mengawasi kinerja seluruh perusahaan.

2. *CIST Division Head* (Kepala Divisi CIST)

Kepala divisi CIST bertugas untuk melapor langsung kepada Dewan Direksi. Kepala divisi ini bertanggung jawab atas seluruh kegiatan operasional dan strategi dalam divisinya.

3. *Data Platform Department Head* (Kepala Departemen Platform Data)

Kepala Departemen Platform Data bertanggung jawab spesifik pada departemen yang mengelola platform data perusahaan.

4. *Business Intelligence Spv* (Supervisor Business Intelligence)

Supervisor *Business Intelligence* fokus mengawasi tim dan proyek yang berkaitan dengan *Business Intelligence* (BI), seperti analisis data, pembuatan laporan, dan *dashboard*.

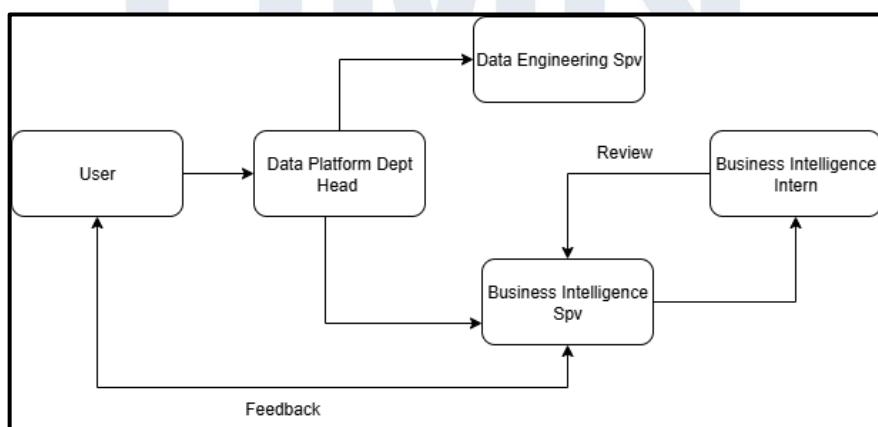
5. *Business Intelligence Engineer*

Merupakan posisi pelaksana sekaligus posisi magang yang berada di bawah pengawasan langsung Supervisor *Business Intelligence*. Seorang *Business Intelligence Engineer* memiliki tugas untuk mengelola infrastruktur data, melakukan analisis data, serta membuat alat dan laporan BI untuk mendukung pengambilan keputusan bisnis.

Dalam struktur organisasi ini, kegiatan magang dilakukan dengan posisi sebagai *Business Intelligence Intern* pada *Data Platform Department*. Penugasan yang diberikan berupa pembuatan dashboard untuk keperluan karyawan SERA dan anak perusahaannya, serta melakukan *user requirements meeting* untuk mengetahui kebutuhan *user* sebelum pengembangan *dashboard* dilakukan.

### 3.1.2 Koordinasi

Gambar 3.2 merupakan diagram mengenai alur koordinasi pekerjaan yang dilakukan selama kegiatan magang di SERA. Pada tahap awal, user melakukan request pembuatan *dashboard* kepada pengawas kegiatan magang, selaku Kepala Departemen *Data Platform* yang dipegang oleh Ibu Farika Maharani. Hal ini merupakan kegiatan dimana *user* menyampaikan permohonan pembuatan *dashboard* Power BI untuk mendukung kegiatan keputusan bisnis mereka. Setelah itu, informasi tersebut kemudian disalurkan kepada pembimbing kegiatan magang, selaku *Supervisor Business Intelligence* yang dipegang oleh Ibu Daulika Pratiwi. Informasi tersebut juga disalurkan kepada *Supervisor Data Engineering*.



Gambar 3.2 Bagan Alur Koordinasi

Sumber: [www.sera.astra.co.id](http://www.sera.astra.co.id) [11]

Setelah informasi mengenai request pengembangan *dashboard* tersebut disalurkan kepada dua pihak, user dengan tim data platform pun melakukan meeting dimana mereka menjelaskan keperluan bisnis mereka dan gambaran *dashboard* yang mereka inginkan. Pembimbing (*Supervisor Business Intelligence*) kemudian merumuskan rangkuman tugas yang jelas dan mendelegasikan tugas tersebut terhadap posisi kegiatan magang sebagai *Business Intelligence Intern* untuk dikerjakan. Setelah selesai, proses penggeraan tugas dilanjutkan dengan kegiatan kerja sama dengan tim *Data Engineering*. Setelah menyelesaikan tugas tersebut, akan diserahkan hasilnya kembali kepada Pembimbing untuk ditinjau, dan jika diperlukan, akan melakukan revisi berdasarkan saran yang diberikan, membentuk siklus kerja yang efektif untuk pembelajaran dan penjaminan kualitas.

### 3.2 Tugas yang Dilakukan

Tabel 3.1 berisi rincian detail pekerjaan yang dilakukan selama kegiatan magang. Terdapat empat proyek yang dikerjakan, yaitu :

Tabel 3.1 Detail Pekerjaan yang Dilakukan

No.	Job Desk	Kegiatan	Durasi
1.	Pengadaan Self Service Training Power BI		
1.1	Memberikan dukungan dan pelatihan pada pengguna akhir	4 Agustus 2025 – 22 Agustus 2025	
2.	Pembuatan Dashboard E-Procurement		
2.1	Mengembangkan dan memelihara dashboard interaktif menggunakan Power BI	25 Agustus 2025 – 29 Agustus 2025	
2.2	Melakukan transformasi data, DAX dan visualisasi efektif	1 September 2025- 10 Oktober 2025	
2.3	Memastikan akurasi dan integritas data yang digunakan	24 November 2025 – 31 Desember 2025	

3.	Pembuatan Dashboard Monitoring Stock Integration	
3.1	Mengembangkan dan memelihara dashboard interaktif menggunakan Power BI	13 Oktober 2025 – 24 Oktober 2025
3.2	Melakukan transformasi data, DAX, dan visualisasi efektif	24 Oktober 2025 – 7 November 2025
3.3	Memastikan akurasi dan integritas data yang digunakan	10 November 2025 – 21 November 2025
4.	Metabase Sharing Session	
4.1	Melakukan kueri basis data dan ekstraksi data menggunakan SQL	13 Oktober 2025 – 31 Oktober 2025
4.2	Memberikan dukungan dan pelatihan kepada pengguna akhir	3 November 2025

### 3.3 Uraian Pelaksanaan Kerja

#### 3.3.1 Proses Pelaksanaan

Bagian ini berisi rincian pekerjaan yang dilakukan dalam kegiatan magang selama bekerja di perusahaan. Tugas pengkerjaan yang dilakukan berupa training serta pengembangan *dashboard*. Hal ini akan dijelaskan dengan detail dan akan ditambahkan juga foto-foto proses dan hasil pekerjaan sebagai dokumentasi kegiatan magang. Berikut adalah rinciannya:

##### 3.3.1.1 Pengadaan *Self Service Training* Power BI

Kegiatan ini merupakan sebuah proyek dimana sektor Business Intelligence mengadakan *training self service* Power BI, yaitu sebuah pelatihan bagi karyawan yang ingin belajar lebih banyak mengenai analisis dan visualisasi data menggunakan tools Power BI.

##### 3.3.1.1.1 Memberikan dukungan dan pelatihan pada pengguna akhir

1. Melakukan Diskusi Kerangka Kegiatan

Pada bagian ini, tim *Business Intelligence* melakukan diskusi mengenai apa saja yang diperlukan dalam pelatihan tersebut, mulai dari penentuan pemateri, penyusunan materi, penetapan jadwal, hingga platform yang akan digunakan. Tujuan utama dari tahap ini adalah untuk menyamakan persepsi dan menetapkan dasar yang kuat bagi seluruh rangkaian kegiatan. Saat kegiatan magang berlangsung, tim data telah membuat semua materi dan kurikulum yang dibutuhkan untuk pelatihan tersebut. Sehingga tugas seorang *business intelligence intern* disini adalah untuk memahami kurikulum yang diberikan agar dapat mendampingi peserta pelatihan dengan baik.

## 2. Melakukan Diskusi dengan Pihak Karyawan SELOG

Tim *Business Intelligence* melakukan koordinasi dan sosialisasi dengan pihak karyawan SELOG yang sudah pernah mengikuti pelatihan tersebut tahun lalu untuk membantu mengatur alur kegiatan dan peserta yang akan berpatisipasi. Hal ini memastikan agar acara dapat berjalan dengan lancar. Diskusi ini dilakukan secara *hybrid* melalui Teams.

## 3. Pelaksanaan Pelatihan

Pelatihan dilakukan di aula untuk menampung 17 peserta. Prosedur dilakukan dengan pembawaan presentasi mengenai pengenalan *Power BI* serta teori – teori yang berkaitan. Setelah itu, kegiatan dilanjutkan dengan praktikum dimana peserta membuat dashboard sederhana berdasarkan instruksi modul pelatihan. Kegiatan ini dilakukan secara bertahap untuk memastikan setiap peserta memahami materi dan tidak ada yang tertinggal dari yang lain. Pengawasan setiap barisan meja peserta dilakukan untuk memastikan tidak ada kendala dalam progres pembuatan *dashboard*.

Dalam praktikum ini, setiap peserta akan diberikan data set penjualan mentah yang relevan dengan bisnis perusahaan. Tugas mereka adalah mengolah dan mengubah data tersebut menjadi sebuah dashboard yang informatif dan interaktif dari awal. Peserta akan mempraktikkan cara membuat berbagai jenis visualisasi, mulai dari grafik tren penjualan berdasarkan waktu, diagram batang untuk membandingkan performa penjualan antar negara, hingga menampilkan *Key Performance Indicator* (KPI) tunggal seperti rata-rata keuntungan (*AverageProfit*). Selain itu, mereka juga akan belajar menyajikan data dalam bentuk tabel detail dan menambahkan fitur fungsional seperti filter tanggal untuk analisis yang dinamis. Latihan praktik ini dirancang untuk memastikan peserta mendapatkan pengalaman langsung dalam menerapkan konsep yang telah dipelajari, sehingga mereka mampu mengubah data mentah menjadi wawasan bisnis yang berguna untuk pengambilan keputusan.

#### 4. Dokumentasi, Pembagian Sertifikasi dan *Doorprize*

Setelah pelatihan selesai, dilakukan foto bersama dan evaluasi untuk mengukur efektivitas program. Evaluasi ini meliputi pengumpulan umpan balik dari peserta melalui kuesioner dan analisis hasil tugas atau post-test. Berdasarkan hasil evaluasi dan partisipasi, peserta yang telah memenuhi kriteria kelulusan akan diberikan sertifikat sebagai tanda penyelesaian pelatihan. Pembagian *doorprize* juga dilakukan bagi satu peserta yang membuat *dashboard* yang paling baik berdasarkan penggunaan visualisasi, kerapian, dan tata letak *layout* serta warna.

##### 3.3.1.2 Perbaikan Dashboard E-Procurement

Kegiatan perbaikan Dashboard *E-Procurement* ini dilakukan sebagai tindak lanjut request dari tim procurement. Latar belakang dari

perubahan ini adalah adanya keinginan menyesuaikan visualisasi dashboard dengan kebutuhan proses bisnis miliki tim procurement saat ini. Kegunannya adalah untuk mendapatkan wawasan mengenai vendor yang paling *cost-effective* untuk perusahaan serta untuk memonitor progress tender terkini.

### **3.3.1.2.1 Mengembangkan dan Memelihara Dashboard Interaktif Menggunakan Power BI**

#### **1. Pengarahan dan Penjelasan Formulir CR Oleh *Supervisor***

Sebagai tahap awal dari keseluruhan proses, sebuah sesi pengarahan diselenggarakan dan dipimpin langsung oleh Supervisor. Dalam sesi ini, Supervisor memberikan penjelasan yang mendetail mengenai setiap poin perubahan visualisasi yang akan diimplementasikan. Seluruh acuan perubahan tersebut didasarkan pada formulir *Change Request* (CR), sebuah dokumen formal yang telah disusun secara sistematis setelah dilakukannya sesi diskusi dan pengumpulan kebutuhan (*requirement gathering*) bersama dengan tim *procurement* selaku pengguna akhir (*user*) dari dashboard ini. Tujuan dari sesi pengarahan ini adalah untuk menyamakan persepsi dan memastikan bahwa tim teknis memiliki pemahaman yang komprehensif terhadap ekspektasi dan tujuan akhir dari setiap item perubahan yang diminta.

#### **3.3.1.2.2 Melakukan Transformasi Data, DAX dan Visualisasi Efektif**

##### **1. Implementasi Perbaikan Bar Chart Partisipasi Tender**

Setelah diberi akses dari tim teknis untuk mendapatkan database yang digunakan, proses perbaikan dimulai dengan

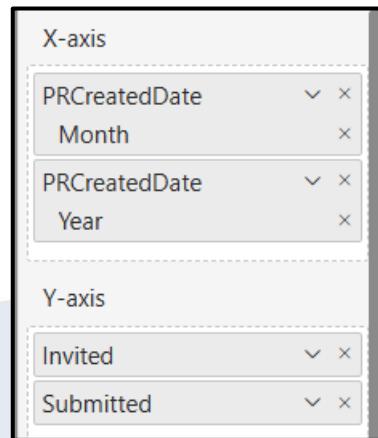
perubahan pada *bar chart tender participant*. Metodologi tender Participant yang terbaru dihitung berdasarkan jumlah rata-rata per bulan vendor yang sudah di invite dan melakukan submission.



Gambar 3.3 Measure Agreggrasi Nilai Rata-Rata

*Measure* pada Gambar 3.3 merupakan *measure* yang digunakan untuk rata-rata vendor yang di *invite*. Kalkulasi untuk vendor yang sudah mengirimkan dokumen juga mirip hanya saja bagian value “*submitted*” tidak dihitung” alasan logika ini digunakan dikarenakan user ingin melihat nilai rata- rata vendor yang diundang dan yang telah mengirimkan dokumen. Sedangkan alasan mengapa *measure average invited* juga menghitung value “*submitted*” dikarenakan user ingin mengetahui berapa banyak vendor yang lanjut ke tahap “*submitted*” dari tahap “*invited*”.

Perhitungan dimulai dengan mendeklarasikan sebuah variabel bernama "vendor". Variabel ini menggunakan fungsi CALCULATE untuk menghitung total jumlah "VendorBranchId" dari tabel performa vendor, namun dengan filter khusus. Filter tersebut memastikan bahwa hanya vendor yang memiliki status "Submitted" atau "Invited" yang akan dihitung. Variabel kedua, "prnumber", kemudian dideklarasikan untuk menghitung jumlah unik atau DISTINCTCOUNT dari "PRNumber" (*Purchase Requisition Number*) dalam tabel yang sama, sesuai dengan konteks filter yang berlaku. Akhirnya, measure ini mengembalikan hasil berupa pembagian antara variabel "vendon" dengan variabel "prnumber". Ini secara efektif memberikan nilai rata-rata vendor yang diundang atau berpartisipasi untuk setiap PR unik yang ada.



Gambar 3.4 Pengaturan Visualisasi Bar Chart Agregasi Rata-Rata

Gambar 3.4 menampilkan pengaturan visualisasi untuk bar chart tersebut. X-axis berisi bulan dan tahun berdasarkan PRCreatedDate untuk mengetahui jumlah rata-rata partisipasi tender setiap bulannya. Y-axis berisi *measure* yang dibuat dengan cara mengkalkulasi jumlah *vendor* yang diundang dan yang sudah mengirimkan penawaran berdasarkan jumlah *distinct vendor*. Tampilan akhir dari perbaikan bar chart tersebut menampilkan dua status: jumlah rata-rata *vendor* yang telah diundang dan jumlah rata-rata *vendor* yang telah mengirimkan penawaran. Data ini diambil dari tabel yang berisi data performa *vendor*.

## 2. Melakukan Perbaikan *Bar Chart Tender Progress*

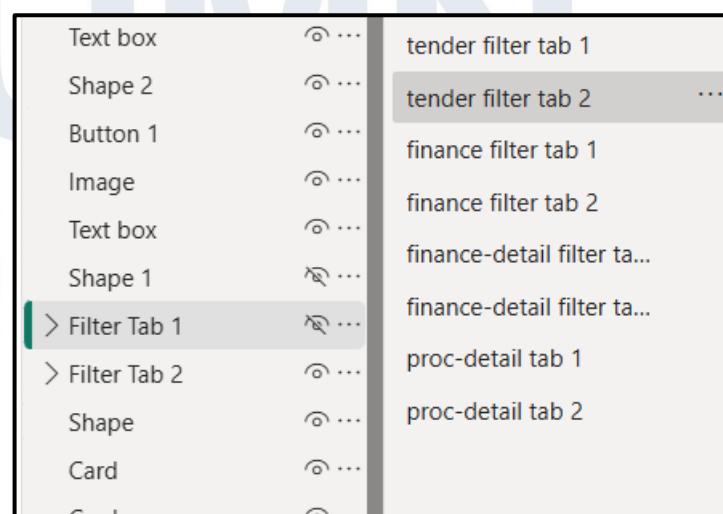
Perubahan Bar Chart Tender Progress diganti menjadi Funnel Chart dilakukan agar dapat menampilkan visualisasi yang lebih jelas mengenai progress *vendor* dalam mengikuti tahap-tahapan tender. Dalam Funnel Chart tersebut terdapat y-axis yang berisi status proses PR, status tender sedang diproses, status tender yang menang tender, status tender yang sudah lewat dari waktu SLA (*Service Level Agreement*), status kontrak yang sedang diproses, status *vendor* menunggu pembayaran dan status *vendor* yang

sudah dibayar. Semua status ini diambil dari kolom status pembelian barang dengan kuantitas sebagai metriknya.

### 3. Melakukan Perbaikan *Slicer*

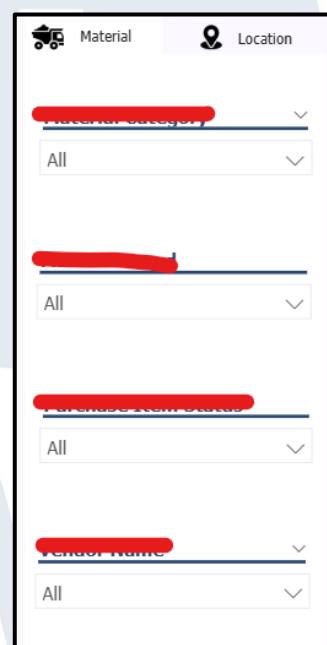
Penyempurnaan fungsionalitas Slicer diimplementasikan agar pengguna dapat melakukan penyaringan (filtering) data visual yang lebih spesifik. Untuk mengatasi tantangan antarmuka di mana terdapat terlalu banyak jenis visualisasi slicer dalam satu halaman dashboard, struktur filter diorganisasi ulang menjadi dua tab utama: tab "Material" dan tab "Lokasi". Pengelompokan ini dilakukan agar semua filter dapat dimuat dalam satu halaman tanpa membuatnya terlihat padat, sehingga meningkatkan kemudahan navigasi bagi pengguna.

Proses teknis pembuatan filter multi-tab ini dimulai dengan mengelompokkan (*grouping*) setiap visualisasi slicer ke dalam dua grup seleksi (*selection group*). Grup pertama berisi semua slicer yang terkait dengan material, dan grup kedua berisi semua slicer yang terkait dengan lokasi.



Gambar 3.5 Penggunaan Fitur Bookmark untuk Membuat Slicers Tab

Setelah itu, fitur "*Bookmark*" dimanfaatkan untuk mengontrol visibilitas setiap grup. Gambar 3.5 menunjukan cuplikan bagian *Selection view* dan *Bookmark*. Dibuat dua *bookmark* setiap *page*: "*Tender Filter Tab 1*" yang diatur agar hanya menampilkan grup *slicer material* (*Selection Filter Tab 1*), dan "*Tender Filter Tab 2*" yang diatur agar hanya menampilkan grup *slicer lokasi* (*Selection Filter Tab 2*). Sebagai langkah akhir, kedua bookmark ini dihubungkan dengan tombol (*button*) yang sesuai.



Gambar 3.6 Tampilan Akhir Interactive Slicers Tab

Gambar 3.6 merupakan hasil akhir dari perbaikan. Ketika pengguna mengklik tombol "*Material*", sistem akan mengaktifkan *bookmark* pertama dan menampilkan kumpulan *slicer material*. Sebaliknya, jika tombol "*Lokasi*" diklik, *bookmark* kedua akan aktif dan menampilkan kumpulan *slicer lokasi*, menciptakan sistem navigasi tab yang efektif.

## 6. Menambahkan *Card* Biaya Total Sebenarnya

Sebuah *Card* yang menampilkan biaya total sebenarnya ditambahkan untuk *dashboard* tersebut. Penambahan komponen visual ini memiliki tujuan untuk menyediakan visibilitas langsung terhadap total biaya sebenarnya yang telah dikeluarkan perusahaan untuk setiap aktivitas pembelian pasokan. Dengan adanya kartu ini, para pemangku kepentingan, seperti manajer pengadaan dan tim keuangan, dapat dengan cepat memperoleh gambaran umum agregat mengenai pengeluaran aktual tanpa perlu melakukan analisis mendalam pada data transaksional yang mentah.

Untuk mengimplementasikan fitur ini, sebuah *measure* (pengukuran) kustom dibuat menggunakan bahasa formula DAX (*Data Analysis Expressions*). Formula DAX ini dirancang untuk secara dinamis menghitung dan menjumlahkan seluruh nilai dari kolom biaya aktual yang relevan di dalam model data. Penggunaan *measure* memastikan bahwa nilai yang ditampilkan pada kartu ini selalu akurat dan akan secara otomatis diperbarui setiap kali data disegarkan atau ketika pengguna menerapkan filter tertentu pada *dashboard*, sehingga memberikan informasi yang *real-time* dan kontekstual.

#### 8. Melakukan Perbaikan *Card Cost Saving*

Kartu indikator "*Cost Saving*" atau Penghematan Biaya juga mengalami perbaikan yang signifikan. Komponen ini krusial karena berfungsi sebagai salah satu Indikator Kinerja Utama (KPI) yang mengukur efektivitas dan efisiensi tim pengadaan. Tujuan utamanya adalah untuk secara kuantitatif menunjukkan seberapa besar penghematan yang berhasil dicapai dari proses tender dan negosiasi dengan membandingkan antara harga anggaran atau harga acuan dengan harga pembelian akhir.

Sama seperti kartu "*Total Cost Actual*", fungsionalitas di balik kartu ini juga ditenagai oleh sebuah *measure* yang kompleks yang dirumuskan menggunakan DAX. Formula DAX yang digunakan tidak hanya sekadar menjumlahkan satu kolom, tetapi melakukan kalkulasi perbandingan, yaitu menghitung selisih antara kolom 'harga anggaran' dan kolom 'harga aktual'. Hasil dari perhitungan ini kemudian diagregasikan untuk menampilkan total penghematan secara keseluruhan. Perbaikan ini memastikan bahwa metrik yang ditampilkan lebih akurat dan dapat diandalkan sebagai bukti nyata dari kontribusi departemen pengadaan terhadap efisiensi finansial perusahaan.

#### 9. Memperbaiki Tabel di *Page Detail*

Upload Kwitansi Date (Paid)	Upload SPK Date (Work In Progress)

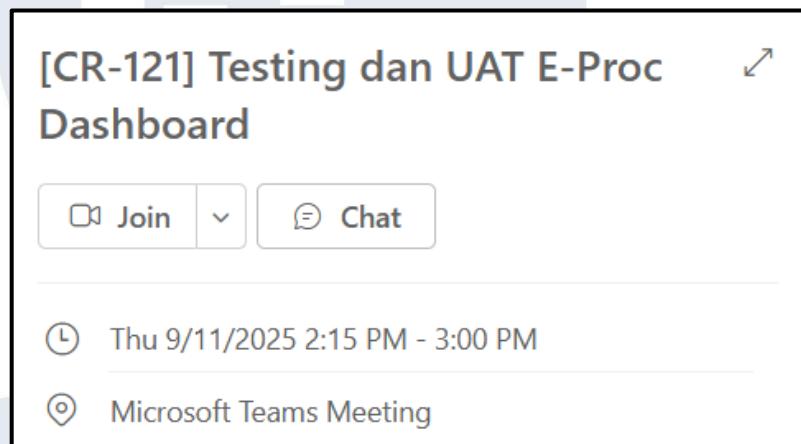
Gambar 3.7 Tampilan Kolom Upload Kwitansi Date dan Upload SPK Date

Berdasarkan Gambar 3.7, perbaikan tabel di *Page Detail* meliputi standardisasi format penanggalan setiap kolom, penambahan kolom tanggal *upload spk date* untuk status pembelian barang vendor yang masih *work in progress*, penambahan kolom tanggal *upload spk date* untuk status pembelian barang yang sudah dibayar serta penambahan kolom

*description* untuk menjelaskan alasan *vendor* tidak lulus proses tender. Semua ini menggunakan DAX function SELECTEDVALUE(). Fungsi ini berguna untuk mengembalikan nilai *distinct* yang berdasarkan *filter* yang ditentukan. Tipe tanggal yang ditampilkan pun juga diselaraskan agar semua memiliki format yang sama dan mudah dibaca oleh *user*.

### 3.3.1.2.3 Memastikan Akurasi dan Integritas Data yang Digunakan

1. Melakukan UAT (*User Acceptance Testing*)



Gambar 3.8 Jadwal Testing dan UAT E-Proc Dashboard

Setelah seluruh tahap pengembangan dan perbaikan teknis selesai, proyek memasuki fase krusial berikutnya, yaitu User Acceptance Testing (UAT) seperti pada Gambar 3.8. Sesi UAT ini merupakan kegiatan kolaboratif di mana pengguna akhir dari tim pengadaan (user) secara langsung berinteraksi dengan dashboard yang telah diperbarui. Tujuan utama dari fase ini adalah untuk memvalidasi bahwa semua perubahan dan fitur baru telah sesuai dengan kebutuhan bisnis yang telah didokumentasikan dalam Change Request (CR) dan berfungsi

seperti yang diharapkan dalam lingkungan operasional sehari-hari.

Selama proses UAT, pengguna secara aktif menguji setiap aspek dari dashboard. Mereka melakukan pengecekan mendalam terhadap akurasi data pada berbagai visualisasi, mencoba semua fungsionalitas filter untuk memastikan data merespons dengan benar, dan memverifikasi bahwa perbaikan pada tabel detail telah diimplementasikan dengan tepat. Tahap ini merupakan kesempatan vital bagi pengguna untuk memberikan ulasan dan masukan konstruktif. Setiap temuan, baik itu bug, ketidaksesuaian, maupun saran untuk penyempurnaan lebih lanjut, dicatat dengan saksama untuk ditindaklanjuti sebelum dashboard secara resmi diluncurkan untuk penggunaan penuh.

Pada sesi UAT dashboard ini, user secara spesifik meminta penjelasan untuk pemakaian fitur export data > underlying data, perbaikan kalkulasi measure untuk card Cost Saving, dan penambahan slicer baru. Kegiatan magang dilanjutkan dengan memperbaiki fitur tersebut dengan cara meletakkan measure ke dalam tabel asal mereka serta menyesuaikan logika kalkulasi sesuai masukan.

Setelah perbaikan tindak lanjut diselesaikan, user melakukan validasi ulang dan menyatakan sangat puas dengan hasil akhirnya. User memberikan persetujuan resmi (formal sign-off) dan mengonfirmasi bahwa visualisasi baru (seperti Funnel Chart) jauh lebih informatif dibandingkan versi sebelumnya. Dengan demikian, tujuan akhir proyek perbaikan ini dinyatakan tercapai sepenuhnya. Dashboard kini berhasil berfungsi sebagai alat monitoring utama yang andal bagi tim procurement untuk

memantau progres tender secara real-time serta mengidentifikasi vendor yang paling cost-effective bagi perusahaan secara akurat.

## 2. Finalisasi dan *Deployment* ke *Production*

Setelah seluruh feedback yang diberikan oleh pengguna akhir (user) telah dikonfirmasi dan diterima secara menyeluruh, proses pengembangan dapat dilanjutkan ke tahap final. Penerimaan feedback ini menandakan bahwa fungsionalitas dan tampilan dashboard telah divalidasi dan sesuai dengan kebutuhan bisnis yang diharapkan.

Hasil akhir antarmuka dashboard ini terbagi menjadi dua halaman utama yang saling melengkapi. Halaman Overview dirancang untuk menyajikan ringkasan eksekutif e-procurement yang memuat card informasi penghematan biaya, visualisasi dalam bentuk diagram batang, diagram corong, serta daftar *top vendor*. Sedangkan halaman Detail menyajikan data granular berupa tabel lengkap yang berisi data riwayat proses tender. Untuk memastikan kemudahan navigasi dan penyaringan data yang dinamis, kedua halaman tersebut telah dilengkapi dengan fitur tab paginated slicer.

Langkah krusial selanjutnya adalah melakukan deployment dashboard ke lingkungan production. Proses ini melibatkan pemindahan dashboard dari server pengembangan atau pengujian ke server live, sehingga dashboard tersebut resmi aktif dan dapat diakses secara penuh oleh semua pengguna yang berkepentingan. Bersamaan dengan deployment tersebut, mekanisme data syncing juga turut diimplementasikan. Sinkronisasi ini diatur untuk memastikan adanya pembaruan data secara berkala dan otomatis dari sumber data utama, sehingga dashboard tersebut selalu

menyajikan data yang paling terkini dan akurat bagi para pengambil keputusan.

### **3.3.1.3 Pengembangan Dashboard Monitoring Integration 2.0**

Pengembangan *dashboard* ini dilakukan untuk keperluan pemantauan integrasi antar sistem secara komprehensif. Fungsi utama ini memungkinkan analisis yang jauh lebih cepat apabila terjadi permasalahan pada data integrasi. Tidak hanya itu, *dashboard* ini juga dirancang untuk menjadi alat pendukung utama dalam proses rekonsiliasi data, sehingga dapat mempercepat proses tersebut secara signifikan. Tujuan penting lainnya adalah untuk meminimalisir terjadinya gap data; hal ini dicapai dengan meningkatkan kesadaran PIC terkait melalui tools yang disediakan, yang sekaligus bersifat preventif terhadap potensi gap data di masa mendatang. Pada akhirnya, dengan data yang lebih akurat dan terpantau, *dashboard* ini berkontribusi mencegah terjadinya VOC (*Voice of Customer*) terkait reservasi *order customer* yang disebabkan oleh ketidakakuratan data stok.

#### **3.3.1.3.1 Pengembangan Dashboard Monitoring Integration**

##### **1. Melakukan *User Requirements Meeting***

Tahap ini merupakan fondasi krusial dari seluruh proses pengembangan. Kegiatan meeting di microsoft teams ini tidak hanya sebatas mendengar, tetapi secara aktif menggali, mendiskusikan, dan mendefinisikan kebutuhan (*requirements*) spesifik dari tim *Product Development* selaku user utama. *User* mempresentasikan gambaran umum dan ekspektasi mereka terhadap *dashboard*, seringkali menggunakan *spreadsheet* Excel sebagai mock-up atau prototipe visual sederhana.

*Mock-up* ini berfungsi sebagai panduan awal untuk menunjukkan data apa yang ingin mereka lihat dan bagaimana kira-kira tampilan yang diharapkan untuk mempermudah analisis. Kehadiran tim *Business Intelligence* (BI) dalam *meeting* ini sangat penting untuk menerjemahkan kebutuhan bisnis tersebut ke dalam metrik, *Key Performance Indicators* (KPI), dan desain visualisasi yang efektif. Sementara itu, keterlibatan *Data Engineer* sejak awal bertujuan untuk memvalidasi ketersediaan data, mengidentifikasi sistem sumber (FMS, SAP, dll.), dan merencanakan arsitektur data serta proses ETL (*Extract, Transform, Load*) yang diperlukan untuk menyajikan data di *dashboard* secara akurat dan efisien.

## 2. Pembuatan Formulir CR

Setelah kebutuhan user dipahami, langkah selanjutnya adalah formalisasi ke dalam sebuah dokumen *Change Request*. Formulir ini berfungsi sebagai dokumen teknis dan 'kontrak' kesepakatan antara tim IT dengan *user*. Dokumen ini sangat penting untuk memastikan tata; kelola proyek berjalan baik. Formulir ini merinci secara komprehensif rencana pengembangan, dan yang terpenting, menjabarkan '*business logic*' atau '*logic*' analisis data. Ini mencakup definisi metrik (misalnya, bagaimana 'selisih stok' dihitung), aturan pemfilteran data, dan sumber data spesifik yang akan digunakan.

Proses penandatanganan oleh *user*, kepala departemen terkait, *supervisor*, serta kepala divisi IT adalah mekanisme esensial untuk memastikan semua pemangku kepentingan setuju dengan ruang lingkup pekerjaan, sehingga menghindari perubahan permintaan di tengah jalan. Sesuai CR tersebut, *dashboard* akan memiliki dua page utama:

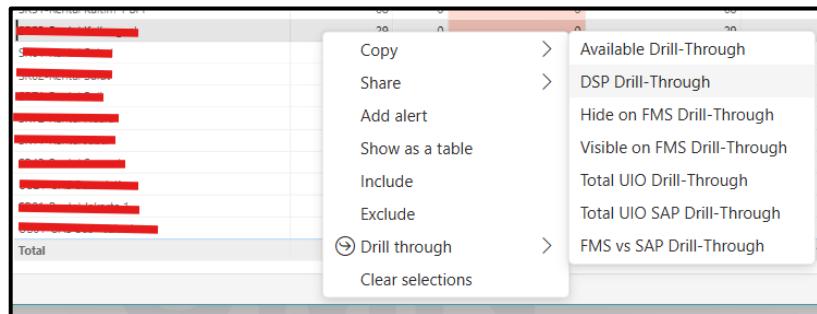
a) *Page Summary*: Dirancang untuk memberikan pandangan 'helikopter' atau ringkasan eksekutif. Ini menampilkan KPI utama seperti total unit disposal, unit yang tersedia, dan metrik paling kritis: jumlah total selisih stok antara sistem FMS dan SAP.

b) *Page Detail*: Berfungsi sebagai level analisis mendalam untuk menginvestigasi temuan di Page Summary. Page ini akan menampilkan data granular, seperti status spesifik kendaraan (*DSP, Deleted, Available*), nomor plat kendaraan, lokasi cabang (*branch*), dan atribut *detail* lainnya yang diperlukan untuk proses rekonsiliasi.

### **3.3.1.3.2 Melakukan Transformasi Data, DAX (Data Analysis Expression), dan Visualisasi Efektif**

#### **1. Pengembangan *Page Summary***

Tahap ini adalah eksekusi teknis dari desain *Page Summary* yang telah disepakati di CR. Kegiatan magang dimulai dengan membangun visualisasi menggunakan *tools* Power BI. 'Slicer Branch' dibuat agar user dapat dengan mudah memfilter keseluruhan data di page berdasarkan lokasi atau cabang tertentu. 'Card Latest Data Sync' dan 'Current Date' ditambahkan untuk memberikan informasi krusial mengenai aktualitas data (data freshness). 'Latest Data Sync' menggunakan formula DAX (*Data Analysis Expressions*) yang disesuaikan ke zona waktu lokal (UTC + 7 atau WIB) untuk menunjukkan kapan data terakhir kali berhasil diperbarui dari sistem sumber.



Gambar 3.9 Tampilan Penggunaan Drill-Through

Fokus utama *page* ini adalah '*Tabel Summary*' yang menampilkan KPI inti: jumlah kendaraan tersedia, kendaraan berstatus disposal, stok tidak valid, dan selisih stok integrasi FMS vs. SAP. Fitur '*drill-through*' yang seperti yang dilihat pada Gambar 3.9 diimplementasikan pada setiap kolom metrik ini. Fitur ini memungkinkan user untuk mengklik sebuah angka (misalnya, angka 'selisih stok' di cabang X) dan langsung berpindah ke *Page Detail* untuk melihat rincian unit apa saja yang membentuk angka selisih tersebut, sehingga analisis menjadi sangat cepat.

#### 4. Pengembangan *Page Detail*

Pengembangan *page* ini difokuskan untuk menyediakan data granular yang diperlukan untuk investigasi dan rekonsiliasi, sesuai tujuan awal *dashboard*. Sesuai spesifikasi CR, *page* ini berisi sebuah tabel atau matriks mendetail. Kolom-kolom yang ditampilkan mencakup semua atribut yang diperlukan untuk identifikasi unit, seperti nomor plat mobil, *vehicle identification number* (VIN), status kendaraan di FMS (DSP, Deleted, Available), padanan status di SAP, dan lokasi cabang.

*Page* ini dirancang untuk menerima filter konteks dari *Page Summary* melalui fitur *drill-through*. Sebagai contoh, jika user mengklik kolom selisih stok di cabang Jakarta pada *Page*

*Summary*, makaan *drill-through* mengarahkan ke page duplikat detail yang akan otomatis terfilter dan hanya menampilkan unit-unit bermasalah di cabang Jakarta, sehingga PIC terkait dapat langsung melakukan investigasi unit-per-unit tanpa perlu mencari data secara manual lagi.

### 3.3.1.3.3 Memastikan Akurasi dan Integritas Data yang Digunakan

#### 1. Validasi Data

Langkah ini merupakan fase quality check internal yang krusial yang dilakukan oleh tim *Data Platform* sebelum dashboard diserahkan kepada *user* untuk *User Acceptance Testing* (UAT). Tujuannya adalah untuk memastikan integritas dan kebersihan data paling fundamental, sehingga sesi UAT dapat berjalan efisien dan fokus pada validasi *business logic* serta fungsionalitas, bukan pada penemuan kesalahan data dasar.

LicensePlate	Total Duplicate Rows	Total Duplicate EquipmentNumbers
[REDACTED]	1	0
[REDACTED]	1	1
[REDACTED]	1	1
AT03ECC	0	0
[REDACTED]	0	0
[REDACTED]	0	0
<b>Total</b>	<b>3</b>	<b>2</b>

Gambar 3.10 Validasi Data untuk Memastikan Tidak Ada yang Duplikat

Berdasarkan Gambar 3.10, fokus utama dari validasi ini adalah pada kolom-kolom yang seharusnya berfungsi sebagai pengidentifikasi unik untuk setiap asset, yaitu data plat mobil dan equipment number (nomor perlengkapan/aset). Dalam konteks sistem seperti FMS (*Fleet Management System*) dan SAP, setiap

kendaraan fisik harus diwakili oleh satu, dan hanya satu, catatan data yang unik.

Proses validasi ini melibatkan pengecekan data secara teknis pada data model atau tabel akhir yang akan menjadi sumber data dashboard. Tim pengembang akan menjalankan kueri atau skrip khusus untuk menganalisis kedua kolom ini. Metode umumnya adalah dengan menghitung jumlah kemunculan (*count*) untuk setiap plat mobil dan setiap *equipment number*. Jika ditemukan ada plat mobil atau *equipment number* yang muncul lebih dari satu kali, ini menandakan adanya duplikasi data yang serius.

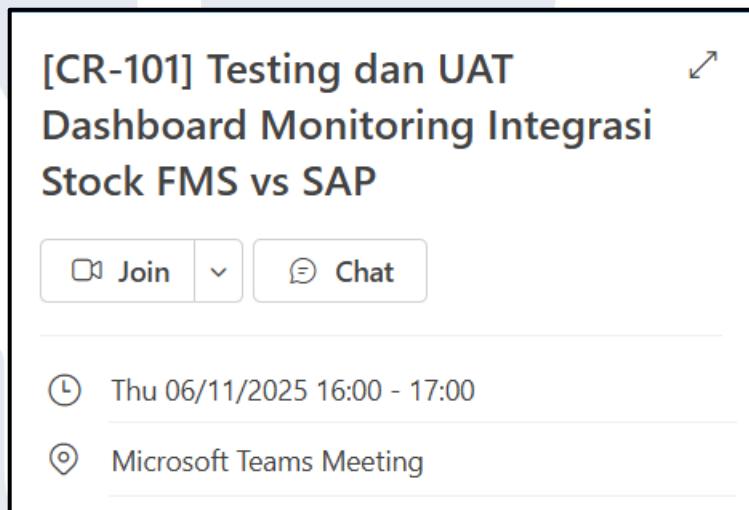
Pentingnya pengecekan duplikasi ini tidak bisa diremehkan. Jika duplikasi data lolos ke *dashboard*, hal itu akan menyebabkan kekacauan analisis secara masif: Pertama, semua angka agregat di *Page Summary* akan menjadi tidak akurat. Sebagai contoh, jika satu mobil dengan status '*Available*' terduplikasi, maka 'Jumlah Kendaraan yang Tersedia' akan menunjukkan angka yang salah (lebih tinggi dari kenyataan). Kedua, perhitungan 'Selisih Stok Integrasi' akan sepenuhnya salah. Duplikasi dapat menggandakan hitungan di satu sisi (misal *FMS*) tetapi tidak di sisi lain (*SAP*), atau sebaliknya, yang akan menciptakan *gap* data yang kedaluwarsa atau menutupi *gap* data yang sebenarnya ada. Ketiga, hal ini akan merusak proses rekonsiliasi di *Page Detail*. Ketika PIC terkait melakukan drill-through untuk menginvestigasi selisih, mereka akan menemukan data ganda yang membingungkan dan tidak dapat ditindaklanjuti.

Oleh karena itu, validasi ini adalah garda pertahanan terakhir tim IT untuk memastikan bahwa data yang akan diuji oleh *user* adalah data yang bersih, logis, dan dapat dipercaya. Jika duplikasi ditemukan pada tahap ini, tim pengembang akan menelusuri akar

masalahnya di proses ETL (*Extract, Transform, Load*) atau di sistem sumber, memperbaikinya, dan memuat ulang data sebelum melanjutkan ke UAT.

## 2. UAT (User Acceptance Testing)

Setelah pengembangan selesai, *dashboard* diserahkan kepada *user* untuk diuji coba secara intensif dalam sesi UAT. Ini adalah fase kritis untuk validasi seperti yang dilihat pada Gambar 3.11. Dalam UAT, user tidak hanya menguji fungsionalitas, tetapi yang lebih penting adalah memvalidasi akurasi data dan kebenaran '*business logic*'.



Gambar 3.11 Testing dan UAT Dashboard Monitoring Integrasi Stok

User akan mengambil beberapa data sampel dan membandingkannya secara manual dengan data di sistem sumber (FMS dan SAP) untuk memastikan angka yang ditampilkan di dashboard akurat. Setiap temuan ketidaksesuaian data, atau permintaan penyempurnaan dicatat dalam UAT *Feedback Form* untuk ditindaklanjuti oleh tim *Data Platform*.

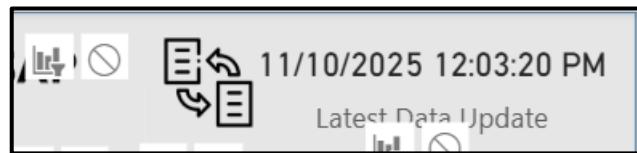
## 3. Penambahan *Count Rows* dan Indeks

Berdasarkan masukan dari sesi UAT, dilakukan penyesuaian fungsionalitas untuk meningkatkan kenyamanan pengguna. Penambahan '*Count Rows*' (jumlah baris) pada Page Detail dan tampilan drill-through sangat penting untuk memberikan konfirmasi kuantitatif kepada user; misalnya, jika Page *Summary* menunjukkan ada 50 selisih, tabel di Page Detail harus menampilkan '*Count Rows*' sebanyak 50, ini memvalidasi bahwa semua data yang relevan telah ditarik.

Penambahan kolom '*Indeks*' atau nomor urut per baris pada tabel detail bertujuan untuk memudahkan referensi dan komunikasi saat proses rekonsiliasi. PIC dapat dengan mudah merujuk ke 'baris indeks nomor 15' saat berdiskusi dengan tim lain untuk menangani data yang bermasalah. Untuk kalkulasi visualisasi tabel, ditambahkan pemakaian *ROWNUMBER()* untuk menghitung baris pada suatu kueri atau tabel sementara secara berurutan. Dia berbeda dengan *RANK()* dimana fungsi tersebut menghasilkan urutan angka yang statis dan tidak akan berganti hasilnya jika kolom nomor tersebut di filter. *ORDER BY()* dan *ASC* digunakan agar kalkulasi mengurutkan berdasarkan suatu kolumn secara virtual dari alfabet A ke Z.

#### 4. Perbaikan Card Latest Data Updated

Permasalahan tampilan integrasi data terbaru diatasi dengan menggunakan fitur '*Edit Interaction*' seperti pada Gambar 3.12 untuk menonaktifkan interaksi (diatur ke '*None*') antara tabel *detail* dan *visual card* tersebut, memastikan *card* tetap menampilkan informasi yang benar terlepas dari apa yang diklik oleh user.



Gambar 3.12 Tampilan Card *Latest Updated*

Kegiatan ini merupakan penyesuaian teknis yang juga umumnya muncul berdasarkan temuan UAT. User menemukan bahwa ketika mereka mengklik satu baris data di tabel pada Page Detail (misalnya, mengklik baris indeks ke-5), visual '*Card Latest Data Updated*' ikut terfilter dan menjadi kosong atau menampilkan data yang tidak relevan. Hal ini tidak diinginkan, karena card tersebut seharusnya bersifat statis (menampilkan waktu update data terakhir dari keseluruhan data) dan tidak terpengaruh oleh interaksi user di tabel.

## 5. Melakukan Laporan Perbaikan Hasil UAT Terhadap User

Setelah seluruh temuan, bug (seperti isu interaksi pada Card *Latest Updated*), dan permintaan penambahan fitur (seperti Count Rows) dari sesi UAT selesai diimplementasikan, langkah selanjutnya adalah melaporkan kembali progres tersebut kepada user. Fase ini bukan sekadar formalitas, melainkan validasi krusial untuk memastikan bahwa solusi teknis yang dibangun benar-benar menyelesaikan masalah bisnis.:

a) Hasil Feedback User: Dalam sesi presentasi akhir, pihak user (Tim Product Development) melakukan tinjauan ulang terhadap dashboard. Hasilnya, user menyatakan sangat puas dengan perbaikan yang dilakukan. Mereka secara spesifik mengapresiasi responsivitas tim data dalam memperbaiki isu duplikasi data dan interaksi filter yang sebelumnya membingungkan. User mengonfirmasi bahwa data yang

ditampilkan kini valid, konsisten antara FMS dan SAP, dan fitur navigasi (drill-through) berfungsi mulus sesuai alur kerja operasional mereka. Dengan hasil ini, user memberikan persetujuan resmi (formal sign-off) bahwa dashboard siap digunakan.

b) Pencapaian Tujuan Akhir Proyek: Dengan disetujuinya dashboard ini, tujuan utama proyek Monitoring Stock Integration 2.0 dinyatakan tercapai sepenuhnya. Waktu yang dibutuhkan PIC untuk menginvestigasi selisih stok berkurang secara signifikan. PIC tidak perlu lagi menarik data manual dari dua sistem berbeda, cukup menggunakan fitur drill-through untuk melihat detail unit yang bermasalah dalam hitungan detik. Dengan data stok yang akurat, risiko tim sales menjanjikan unit yang sebenarnya tidak ada (*ghost stock*) dapat dimitigasi, sehingga mencegah terjadinya komplain pelanggan atau pembatalan order di kemudian hari..

## 5. Finalisasi dan Deployment ke Production

Setelah mendapatkan persetujuan resmi dari pengguna, dashboard siap untuk dipindahkan dari lingkungan pengembangan ke lingkungan produksi. Visualisasi akhir pada halaman Summary difokuskan pada penyajian tabel komparasi mendetail antara jumlah unit kendaraan yang tercatat di sistem SAP dan basis data pembanding. Guna mendukung pemantauan anomali secara cepat, halaman ini dilengkapi dengan kartu informasi waktu pembaruan data terakhir serta kartu indikator yang menonjolkan total selisih data kendaraan yang tidak sinkron di antara kedua basis data tersebut.

Fitur interaktif drill-through juga telah diimplementasikan sebagai mekanisme navigasi antarhalaman. Apabila pengguna

menelusuri nilai spesifik pada halaman Summary, sistem secara otomatis akan mengarahkan pengguna ke halaman Detail yang telah terfilter sesuai konteks nilai yang dipilih sebelumnya. Pada halaman Detail, informasi disajikan lebih mendalam melalui indikator jumlah kendaraan aktual dan status pembaruan data. Eksplorasi data pada halaman ini didukung oleh fitur paginated slicers yang dikelompokkan ke dalam tab Lokasi dan Transportasi.

Tahap selanjutnya adalah proses publikasi yang melibatkan penempatan berkas dashboard ke layanan Power BI Service. Tujuannya adalah memperluas aksesibilitas dashboard bagi pengguna yang berkepentingan (seperti PIC cabang dan manajer) melalui peramban (browser) atau aplikasi seluler, dengan hak akses yang terkendali. Komponen krusial dari proses ini adalah konfigurasi Scheduled Refresh untuk menjamin data selalu terbarui sesuai frekuensi yang disepakati. Hal ini memastikan dashboard berfungsi optimal sebagai alat pemantauan yang komprehensif dan preventif..

### **3.3.1.4 Sharing Session Metabase**

Kegiatan magang ini dilakukan tidak hanya untuk menyelesaikan tugas-tugas teknis, tetapi juga memiliki tujuan strategis yang lebih luas, yaitu untuk menambah wawasan dan memperkaya ilmu pengetahuan tim data *platform* internal. Tim ini, yang sehari-harinya telah sangat mahir menggunakan Power BI, perlu diekspos terhadap berbagai macam alternatif tools Business Intelligence yang ada di *market*. Tujuannya adalah agar tim dapat melakukan evaluasi komparatif, memahami kelebihan dan kekurangan setiap platform, dan pada akhirnya mampu merekomendasikan solusi yang paling tepat guna (*fit-for-purpose*)

untuk berbagai skenario bisnis yang berbeda, yang mungkin tidak selalu ideal jika ditangani hanya dengan Power BI.

### 3.3.1.4.1 Melakukan Kueri Basis Data dan Ekstraksi Data Menggunakan SQL

#### 1. Membangun Relasi Skema Data

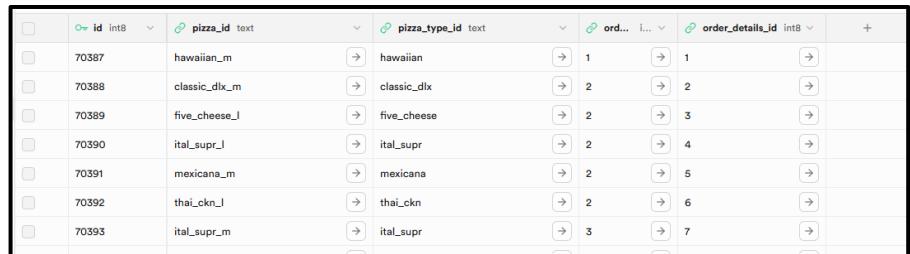
Langkah fundamental pertama adalah menyusun dan memahami relasi antar tabel pada basis data mentah. Berdasarkan gambar pertama, terlihat struktur skema yang terdiri dari beberapa tabel terpisah, yaitu orders, order details, pizzas, dan pizza types. Pada tahap ini, seperti Gambar 3.13 dilakukan identifikasi kunci penghubung atau foreign key (seperti pizza id dan pizza type id) untuk memetakan bagaimana satu data berhubungan dengan data lainnya.

```
36
37
38 INSERT INTO master_data (order_details_id, order_id, pizza_id, pizza_type_id)
39 SELECT
40     od.order_details_id,
41     od.order_id,
42     od.pizza_id,
43     p.pizza_type_id
44 FROM
45     order_details AS od
46 JOIN
47     pizzas AS p ON od.pizza_id = p.pizza_id;
```

Gambar 3.13 Pembuatan Star Schema dengan SQL

Setelah alur relasi dipahami, proses dilanjutkan dengan teknis penggabungan data seperti yang ditunjukkan pada gambar kedua. Menggunakan perintah SQL (INSERT INTO master data... SELECT... JOIN...), data dari tabel-tabel yang terpisah tersebut diekstraksi dan digabungkan (denormalisasi) menjadi satu tabel tujuan bernama master data seperti pada Gambar 3.14 . Proses ini krusial untuk menyederhanakan struktur data, sehingga saat masuk ke tahap visualisasi, performa kueri menjadi lebih

cepat dan pengguna tidak perlu melakukan join tabel yang rumit secara berulang-ulang.



	id	int8	pizza_id	text	pizza_type_id	text	order_id	int8	order_details_id	int8	+
	70387		hawaiian_m	hawaiian	1	1	1				
	70388		classic_dlx_m	classic_dlx	2	2	2				
	70389		five_cheese_l	five_cheese	2	2	3				
	70390		ital_supr_l	ital_supr	2	2	4				
	70391		mexicana_m	mexicana	2	2	5				
	70392		thai_ckn_l	thai_ckn	2	2	6				
	70393		ital_supr_m	ital_supr	3	3	7				

Gambar 3.14 Tampilan Output Berupa Tabel Master Data

## 2. Menghubungkan Database ke Metabase

Setelah tabel master terbentuk, langkah selanjutnya adalah menghubungkan basis data tersebut ke platform Metabase agar dapat diolah secara visual. Pada studi kasus ini, basis data di-hosting menggunakan layanan Supabase. Karena Supabase dibangun di atas arsitektur PostgreSQL, konfigurasi di Metabase dilakukan dengan memilih tipe basis data PostgreSQL pada menu pengaturan Admin, kemudian memilih opsi Add Database.

Praktikan kemudian menginput detail kredensial koneksi yang didapatkan dari pengaturan proyek di dashboard Supabase (bagian Database Connection). Informasi yang dimasukkan meliputi Host (alamat server db.supabase.co), Port (5432), Nama Database (postgres), Username, dan Password. Setelah tombol Save ditekan, Metabase akan melakukan sinkronisasi dan memindai skema data (sync schema), sehingga tabel master data yang telah disiapkan sebelumnya muncul di Metabase dan siap digunakan untuk pembuatan dashboard.

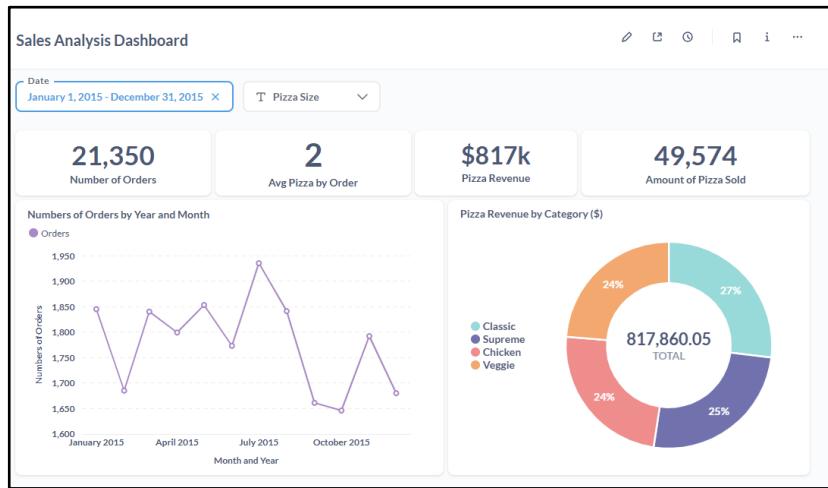
### 3.3.1.4.2 Memberi Dukungan dan Pelatihan kepada Pengguna Akhir

#### 1. Pembuatan Kerangka Presentasi Menggunakan PPT

Langkah fundamental pertama dalam persiapan *Sharing Session* ini adalah perancangan dan pembuatan materi presentasi visual menggunakan *Microsoft Powerpoint*. Proses ini bukan sekadar menyalin fakta, melainkan menyusun alur cerita yang logis dan relevan bagi audiens (tim *data platform*). Isi *slide* presentasi dirancang secara sistematis, dimulai dengan pengenalan umum mengenai *Metabase*, memposisikannya dalam ekosistem tools BI, dan menjelaskan filosofi di baliknya yang berfokus pada kemudahan penggunaan. Bagian selanjutnya masuk lebih dalam secara teknis, menjelaskan arsitektur dasar cara *Metabase* mengambil *query* dan melakukan visualisasi data, sebuah poin krusial untuk membandingkannya dengan cara kerja Power BI (misalnya *Direct Query* vs. *Import* di Power BI). Selain itu, *slide* juga mencakup analisis *pricing plan* *Metabase*, menyoroti opsi open-source gratisnya sebagai alternatif dari model lisensi Power BI. Bagian terpenting adalah penjabaran konsep-konsep utama dari *Metabase* (seperti 'Questions', 'Models', dan 'Pulses') agar tim dapat memetakannya ke konsep serupa di Power BI. Seluruh *slide* ini berfungsi sebagai landasan teoretis sebelum masuk ke bagian praktik melalui live demo.

## 2. Pembuatan Materi Live Demo

Gambar 3.15 merupakan persiapan untuk komponen paling inti dari sharing session. Materi live demo dirancang bukan sebagai studi kasus yang kompleks, melainkan sebagai demonstrasi "zero-to-dashboard" yang menyoroti kecepatan dan intuitivitas *Metabase*. Skenario demo disiapkan secara matang, dimulai dengan menghubungkan ke sumber data yang sudah disiapkan.



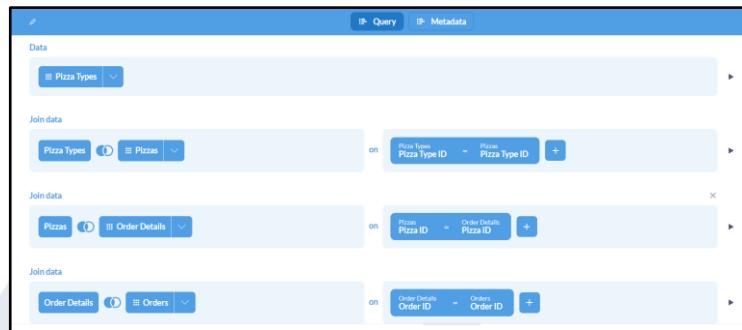
Gambar 3.15 Tampilan Demonstrasi Singkat *Dashboard* Metabase

Kegiatan demo berfokus pada alur kerja pengguna, dimulai dengan pembelajaran cara membuat 'Question' atau pertanyaan dasar. Dari sana, demo menunjukkan cara membuat visualisasi fundamental seperti card (untuk menampilkan KPI tunggal atau 'Metric'), line chart (untuk analisis tren waktu), dan pie chart (untuk perbandingan kategori). Puncak dari demo ini adalah menunjukkan kemudahan dalam merakit visualisasi-visualisasi tersebut menjadi satu dashboard interaktif, dan yang terpenting, cara menambahkan filter atau parameter global (misalnya filter tanggal atau kategori) yang dapat memengaruhi semua bagan di dalam dashboard, sebuah fitur esensial yang pasti akan dibandingkan langsung dengan fungsionalitas serupa di Power BI.

### 3. Pembuatan Materi Live Demo : Create Model

Pada tahap ini, dilakukan pembuatan Model data. Sesuai dengan alur kerja yang dirancang, pembuatan Model ini didahului sebelum membuat Questions. Tujuan dari langkah ini adalah agar proses analisis dan pembuatan Questions di tahap selanjutnya menjadi lebih terorganisir, memiliki standardisasi

data yang jelas, dan untuk meningkatkan performa query seperti pada Gambar 3.16.



Gambar 3.16 Tampilan Pembuatan Model

Terdapat empat langkah dalam pembuatan Model tersebut. Langkah awal dalam penyusunan struktur data ini dimulai dengan mengakses antarmuka utama Metabase dan menavigasi kursor menuju tab menu Model. Pada halaman tersebut, praktikan memulai proses inisialisasi dengan mengklik tombol Add New Model yang terletak pada bagian atas tampilan. Tindakan ini akan membuka sebuah editor visual kosong yang berfungsi sebagai kanvas kerja untuk menyusun logika data. Tahapan ini merupakan gerbang krusial sebelum pengguna dapat melangkah ke proses pemilihan tabel sumber dan pendefinisian relasi antar entitas.

Langkah kedua adalah proses penggabungan data (join) dari beberapa tabel yang relevan. Praktikan memastikan proses join data dilakukan sesuai dengan skema yang tertera pada panduan (gambar), dengan rincian sebagai berikut: Tabel Pizza Types digabungkan dengan tabel Pizzas menggunakan kunci relasi Pizza Type ID. Tabel Pizzas digabungkan dengan tabel Order Details menggunakan kunci relasi Pizza ID. Tabel Order Details digabungkan dengan tabel Orders menggunakan kunci relasi Order ID.

Langkah ketiga adalah penyesuaian metadata. Setelah data berhasil digabungkan, praktikan beralih ke Tab Metadata. Pada tahap ini, dilakukan perubahan tipe data untuk beberapa kolom yang sebelumnya masih terdeteksi sebagai "no semantic" agar lebih sesuai dengan makna datanya: Kolom Ingredient diubah tipenya menjadi Description. Kolom Pizza Price diubah tipenya menjadi Currency. Kolom Orders date diubah tipenya menjadi Join Date. Kolom Orders time diubah tipenya menjadi Join Time.

Langkah terakhir, setelah semua penyesuaian data dan metadata selesai, model tersebut disimpan dengan nama pizza sales questions. Model ini kemudian siap digunakan sebagai dasar untuk pembuatan Questions atau visualisasi data pada tahap berikutnya.

#### 4. Pembuatan Materi Live Demo: *Create Metrics*

Tahap selanjutnya setelah pembuatan model adalah pembuatan *Metric*. Pada bagian ini, praktikan membuat sebuah metrik kustom yang diberi nama "Pizza Revenue". Proses ini diawali dengan memilih sumber data pada bagian "Data" yang akan digunakan. Sesuai alur, sumber data yang dipilih adalah model "pizza modelling" yang telah dipersiapkan pada tahap sebelumnya.

Setelah sumber data ditentukan, praktikan beralih ke bagian "Summarize" untuk membuat kalkulasi. Di sini, praktikan memilih untuk membuat sebuah "custom expression" untuk mendefinisikan "Pizza Revenue".



```
Custom Expression
1 Sum([Quantity] * [Pizza Price])
```

Gambar 3.17 Tampilan Pembuatan Custom Expression

Berdasarkan Gambar 3.17, ekspresi kustom yang dimasukkan adalah  $\text{Sum}([\text{Quantity}] * [\text{Pizza Price}])$ . Formula ini mengalikan jumlah kuantitas pizza yang terjual dengan harga pizza terkait, kemudian menjumlahkan totalnya untuk mendapatkan nilai pendapatan secara keseluruhan.

Tujuan utama dari pembuatan metrik ini adalah untuk reusabilitas. Sebagaimana dijelaskan dalam slide, Metrik "Pizza Revenue" yang sudah disimpan ini dapat digunakan kembali dalam berbagai "questions" atau analisis lain di kemudian hari. Hal ini sangat penting untuk efisiensi kerja karena praktikan tidak perlu menulis ulang formula kustom yang sama secara berulang kali, sekaligus menjaga konsistensi dalam perhitungan.

##### 5. Pembuatan Materi Live Demo : *Create KPI Cards*

Tahapan selanjutnya dalam rangkaian proses ini adalah pengembangan materi untuk demonstrasi langsung yang berfokus pada pembuatan KPI Cards. Pada fase ini, aktivitas utama yang dilakukan adalah *Creating Cards*, sebuah langkah teknis yang bertujuan untuk menghasilkan representasi visual data yang ringkas namun informatif, sebagaimana dicontohkan pada Gambar 3.18 yang menampilkan angka total pesanan atau *Number of Orders*. Proses ini melibatkan penyusunan logika data melalui pembuatan beberapa *Questions* secara individual untuk memunculkan angka tunggal yang akurat, di mana hasil

visualisasi ini memberikan gambaran instan mengenai performa terkini dari variabel yang sedang dianalisis oleh praktikan.

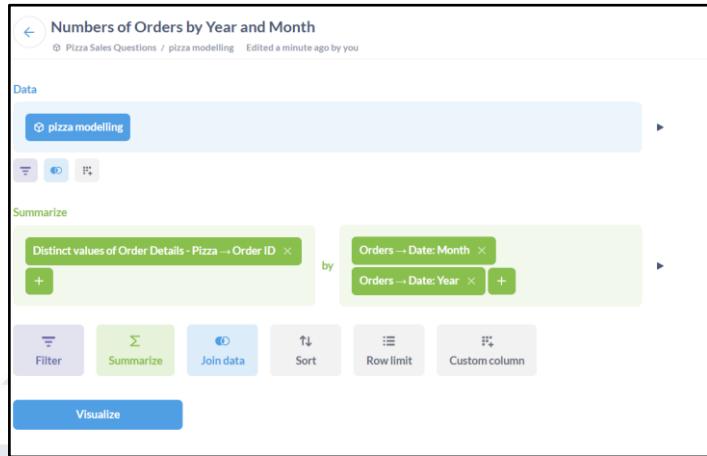


Gambar 3.18 Tampilan Visualisasi Cards

Lebih jauh lagi, langkah pembuatan *Questions* ini memegang peranan strategis dalam konstruksi akhir sistem pelaporan visual. Setiap kartu yang dihasilkan dirancang untuk berfungsi sebagai kartu metrik utama yang menonjolkan indikator vital bagi perusahaan. Nantinya, kumpulan kartu metrik individual ini kemungkinan besar akan diintegrasikan dan disusun ke dalam sebuah dashboard operasional yang lebih komprehensif. Dengan demikian, elemen-elemen visual ini akan menjadi komponen fondasi yang memudahkan pengguna dalam memantau indikator kinerja utama secara cepat dan efisien dalam satu tampilan layar terpadu.

#### 6. Pembuatan Materi *Live Demo : Line Chart*

Tahap berikutnya adalah pembuatan visualisasi data, secara spesifik sebuah *Line Chart* dengan proses pembuatan seperti Gambar 3.19. Tujuan dari visualisasi ini adalah untuk membuat "*Number of Orders by Month and Year*" (Jumlah Pesanan per Bulan dan Tahun) guna menganalisis tren pesanan seiring waktu. Dengan adanya *Line Chart* ini, user dapat dengan mudah membaca tren pesanan.



Gambar 3.19 Tampilan Pembuatan Visualisasi Bar Chart

Proses pembuatannya adalah sebagai berikut. Pertama, pada bagian "Data", praktikan memilih sumber data yang digunakan. Sesuai instruksi, sumber data yang dipilih adalah "pizza modelling", yang merupakan model data terstruktur yang telah disiapkan pada tahap pertama.

Kedua, pada bagian "Summarize", praktikan melakukan agregasi data untuk menyusun grafik. Sesuai dengan penjelasan di slide, Distinct values of Order ID (jumlah nilai unik dari ID Pesanan) digunakan sebagai metrik utama atau sumbu Y-axis. Sedangkan untuk pengelompokan datanya, atau sumbu X-axis, praktikan menggunakan Orders -> Date: Month dan Orders -> Date: Year.

Langkah-langkah ini menghasilkan sebuah visualisasi grafik garis yang menunjukkan pergerakan atau tren jumlah pesanan yang diterima setiap bulan dan tahun, seperti yang terlihat pada gambar di sebelah kanan.

## 7. Pembuatan Materi Live Demo : *Pie Chart*

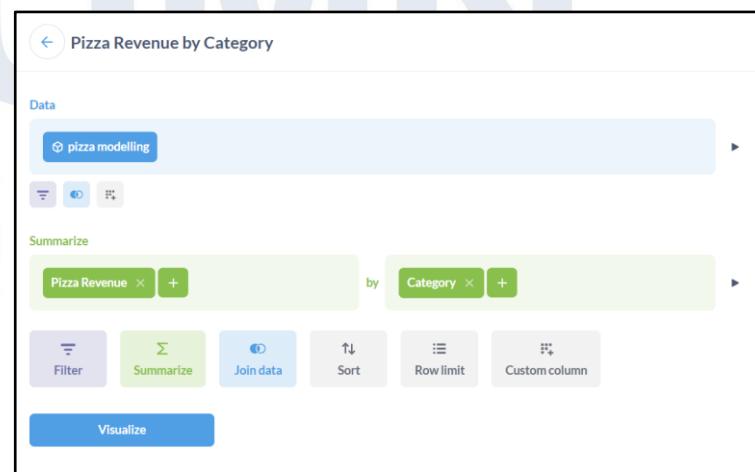
Tahap selanjutnya adalah pembuatan visualisasi berupa Pie Chart. Visualisasi ini dibuat dengan tujuan untuk menganalisis "Pizza Revenue by Category" (Pendapatan Pizza berdasarkan Kategori).

Langkah-langkah yang dilakukan berdasarkan Gambar 3.20 adalah sebagai berikut:

Pertama, pada bagian "Data", praktikan kembali memilih "pizza modelling" sebagai sumber data utama, yang konsisten dengan langkah-langkah sebelumnya.

Kedua, pada bagian "Summarize", praktikan tidak membuat kalkulasi baru. Sebaliknya, praktikan memilih "Metrics", lalu memilih metrik "Pizza Revenue" yang telah dibuat dan disimpan pada tahap kedua. Ini adalah contoh penerapan metrik yang dapat digunakan kembali.

Ketiga, untuk membagi nilai pendapatan tersebut ke dalam beberapa segmen, praktikan menambahkan pengelompokan data (group by) menggunakan kolom "Category".

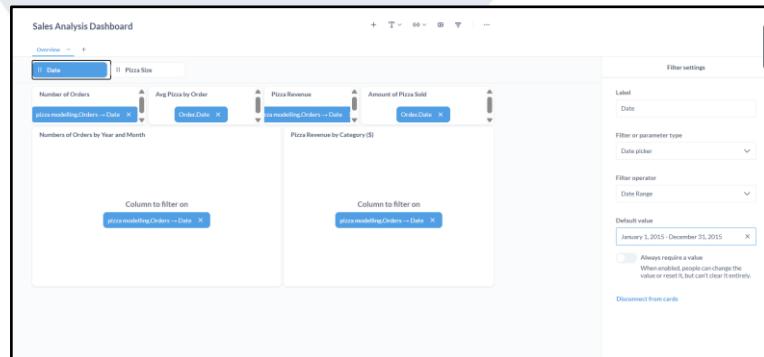


Gambar 3.20 Tampilan Pembuatan Visualisasi Pie Chart

Langkah-langkah ini menghasilkan sebuah grafik lingkaran yang secara efektif menampilkan proporsi atau persentase pendapatan yang disumbang oleh setiap kategori pizza (*Classic, Supreme, Chicken, dan Veggie*), beserta total pendapatan yang dihasilkan.

#### 8. Pembuatan Materi Live Demo : *Filter*

Tahap selanjutnya adalah pembuatan *filter* tanggal. Pada tahap ini, praktikan menambahkan fungsionalitas filter interaktif pada "Sales Analysis Dashboard" yang telah disiapkan. Proses ini dimulai dengan mengklik opsi "*edit dashboard*" untuk masuk ke mode penyuntingan. Setelah itu, praktikan memilih "*Add Filter*" untuk menambahkan filter baru seperti pada Gambar 3.21.



Gambar 3.21 Tampilan Proses Pembuatan Filter

Selanjutnya, praktikan mengkonfigurasi filter tersebut. Filter ini diatur sebagai filter "Date" (Tanggal). Sesuai dengan catatan penting pada slide, filter ini hanya dapat dihubungkan ke visualisasi atau "Questions" yang memiliki kolom data yang relevan untuk difilter.

Kegiatan kemudian dilanjutkan dengan memilih visualisasi mana saja di dalam dashboard yang ingin dihubungkan ke filter tanggal ini. Dari gambar, terlihat filter ini diterapkan pada

beberapa kartu, seperti "Number of Orders" dan "Pizza Revenue by Category", dengan menghubungkannya ke kolom Orders -> Date.

Sebagai langkah terakhir, praktikan mengatur "Default value" untuk filter tersebut. Nilai default ditetapkan untuk rentang tanggal "January 1, 2015 – December 31, 2015". Ini memastikan bahwa saat dashboard pertama kali dibuka, data yang ditampilkan akan secara otomatis berada dalam rentang waktu tersebut, namun pengguna tetap dapat mengubahnya

### 1. Pelaksanaan Sharing Session

Ini adalah momen eksekusi utama dari seluruh rangkaian persiapan. Sesi *sharing* ini dilaksanakan pada jadwal yang telah disepakati bersama tim data platform dan diselenggarakan secara online melalui *Microsoft Teams*. Pelaksanaan dimulai tepat waktu, dibuka dengan pemaparan materi presentasi untuk membangun fondasi pemahaman dan menyamakan persepsi.

Penyampaian materi dilakukan dengan alur yang terstruktur, menjaga agar tetap relevan dengan konteks audiens yang sudah ahli Power BI. Setelah landasan teoretis terbangun, sesi beralih mulus ke live demo. Selama demo berlangsung, penekanan diberikan pada narasi "how-to" secara real-time, sambil secara aktif mengundang interaksi dan menyoroti perbedaan atau persamaan signifikan dalam alur kerja jika dibandingkan dengan Power BI, untuk menjaga agar sesi tetap engage dan relevan dengan kebutuhan wawasan mereka.

### 2. Q&A Session

Sesi tanya jawab dialokasikan secara khusus di akhir acara sebagai bagian krusial untuk mengukur pemahaman dan ketertarikan tim. Bagian ini bukan sekadar formalitas, melainkan menjadi forum diskusi interaktif. Tim data platform, dengan latar belakang Power BI mereka yang kuat, mengajukan pertanyaan-pertanyaan yang sangat tajam dan mendalam. Diskusi tidak hanya berkutat pada fitur yang didemokan, tetapi juga meluas ke area yang lebih teknis seperti batasan data modeling di Metabase (dibandingkan dengan DAX dan *Power Query*), kapabilitasnya dalam menangani join yang kompleks, mekanisme pengaturan keamanan dan hak akses, serta skalabilitas platform untuk menangani *volume* data yang besar. Sesi Q&A ini menjadi sangat produktif dan berhasil memvalidasi tercapainya tujuan awal, yaitu menambah wawasan kritis tim mengenai Metabase sebagai alternatif yang layak dipertimbangkan.

### 3.3.2 Kendala yang Ditemukan

Bagian berikut ini merupakan kumpulan kendala yang ditemukan dalam kegiatan magang selama 6 bulan di PT Serasi Autoraya sebagai *Business Intelligence Engineer*. Berikut merupakan penjabaran kendala yang dialami :

#### 1. Struktur Data dan Tabel Relasional yang Kompleks

Tantangan ini muncul karena data tidak berasal dari satu sistem tunggal, melainkan dari berbagai sumber, terutama pada proyek E-Procurement. Sistem-sistem ini memiliki skema database mereka sendiri yang sudah kompleks, dengan puluhan atau bahkan ratusan tabel. Saat data ini perlu digabungkan untuk analisis, kompleksitasnya berlipat ganda. Tantangannya bukan hanya soal jumlah tabel, tetapi juga dalam memahami '*business logic*' di baliknya; misalnya, bagaimana sebuah status vendor terhubung dengan catatan progressnya di data performa

vendor. Menemukan kunci relasional yang tepat (apakah menggunakan plat mobil, nomor *equipment*, atau ID unik lainnya) dan memahami jenis relasi *database* (*one-to-one*, *one-to-many*) sangat krusial. Kesalahan kecil dalam pemodelan relasi ini dapat mengakibatkan data ganda (duplikasi) atau perhitungan metrik yang salah (misalnya, total stok dan tender yang tidak akurat) di *dashboard*.

## 2. Kebutuhan *Slicers* yang Banyak Membuat Page Dashboard Tidak Memiliki Ruang yang Cukup untuk Menampilkan Semuanya

User memiliki kebutuhan yang valid untuk dapat menganalisis data dari berbagai sudut pandang. Hal ini diterjemahkan menjadi permintaan untuk banyak sekali 'slicers' (filter) di dashboard, seperti filter berdasarkan cabang, status kendaraan, tipe unit, rentang tanggal, status FMS, status SAP, dan lain-lain. Tantangan yang muncul adalah murni terkait desain *interface*. Setiap slicer yang ditempatkan di halaman dashboard akan memakan 'ruang layar' yang berharga. Jika terlalu banyak *slicer* yang ditampilkan secara permanen, ruang untuk menampilkan visualisasi data inti (grafik, tabel, dan KPI card) menjadi sangat sempit, membuat *dashboard* terlihat penuh sesak, tidak rapi, dan sulit dibaca oleh pengguna.

## 3. Kebutuhan User untuk Dashboard yang Mudah Dibaca

Kebutuhan pengguna akan dashboard yang "mudah dibaca" seringkali menjadi sebuah tantangan dalam pengembangan proyek. Ini bukan sekadar kebutuhan, melainkan sebuah kendala desain yang kompleks. Pengguna (*user*) umumnya bukan analis data; mereka adalah manajer atau staf operasional yang membutuhkan jawaban cepat atas pertanyaan bisnis.

Tantangannya terletak pada bagaimana menerjemahkan data yang kompleks, yang berasal dari berbagai sumber, menjadi sebuah antarmuka

visual (UI) yang bersih, tidak mengintimidasi, dan intuitif. Jika *dashboard* yang dihasilkan terlihat penuh sesak, sulit dinavigasi, atau alur pencarian informasinya tidak jelas, pengguna akan mengalami beban kognitif yang tinggi. Dampak akhirnya adalah rendahnya tingkat adopsi (*user adoption*), di mana *dashboard* yang telah dibuat dengan susah payah akhirnya tidak digunakan oleh user.

### 3.3.3 Solusi atas Kendala yang Ditemukan

Bagian berikut berisi penjelasan mengenai solusi atas kendala-kendala yang dijabarkan dalam bagian 3.3.2.

#### 1. Pemeriksaan Validitas Data

Dalam pengembangan *dashboard*, dilakukan validasi data untuk memastikan measure yang dibuat sesuai dengan data yang ada. Proses ini bertujuan untuk menjamin integritas data dan memastikan bahwa measure (formula kalkulasi) yang dibuat telah akurat serta merefleksikan data sumber dengan benar.

Sebagai bagian dari implementasi teknis, sebuah metode praktis diterapkan, yaitu dengan membuat satu halaman (*page*) khusus untuk validasi. Halaman ini diatur agar tersembunyi (*hidden*) dari pengguna akhir dan berfungsi sebagai area pengujian.

Di dalam halaman validasi ini, dilakukan beberapa kegiatan validasi utama :

Pertama, dibuat sebuah measure DAX spesifik untuk memeriksa data stock integration. Pengecekan ini berfokus pada equipment number dan plat kendaraan untuk memastikan tidak terjadi duplikasi data.

Keberhasilan validasi ini sangat penting karena data duplikat dapat menyebabkan kesalahan fatal pada kalkulasi total stok di *dashboard*.

Kedua, dilakukan verifikasi silang (*cross-check*) untuk memastikan kalkulasi yang kompleks, seperti nilai rata-rata pada sebuah visualisasi *bar chart*, telah akurat. Ini dilakukan dengan membandingkan angka yang ditampilkan pada visual tersebut dengan perhitungan manual atau tabel data mentah yang diletakkan di halaman validasi.

## 2. Pembuatan Kumpulan *Slicers* yang Berupa Tab Interaktif.

Solusinya untuk mengatasi kendala banyaknya jumlah *slicers* yang dibutuhkan berfokus pada desain antarmuka pengguna (UI) dan optimalisasi ruang layar. Solusi ini menggunakan bookmark sebagai Teknik utama dalam menyelesaikan permasalahan tersebut.

Hal ini dilakukan dengan pengembangan panel Slicer ‘*Interactive Tabs*’. Ini adalah solusi paling efektif untuk menjaga kebersihan kanvas. Teknik yang sebutkan, yaitu penggunaan 'Bookmark' yang dikombinasikan dengan 'Selection Pane' untuk mengelompokkan (grouping) semua slicer, adalah praktik yang dilakukan selama kegiatan magang. Slicer-slicer tersebut ditempatkan dalam satu grup yang bisa dimunculkan (show) atau disembunyikan (hide) dengan satu tombol (misalnya, ikon 'Filter'), sehingga menciptakan tampilan seperti tab atau panel filter kustom.

## 3. Merancang Desain *Dashboard* Power BI yang Intutif

Untuk mengatasi tantangan keterbacaan tersebut, solusi yang diterapkan adalah perancangan desain yang berpusat pada pengguna (*user-centered design*). Fokusnya adalah memikirkan alur berpikir dan kebutuhan bisnis user, yang diimplementasikan melalui beberapa teknik spesifik:

Pertama, implementasi navigasi yang jelas dan ikonografi yang konsisten. Kejelasan fungsionalitas sangat penting. Pengguna tidak boleh dibiarkan bingung saat navigasi *dashboard*. Oleh karena itu, tombol untuk fungsi kritis dibedakan dengan jelas: tombol untuk kembali ke halaman sebelumnya menggunakan ikon panah (arrow) standar, sementara tombol untuk menghapus semua filter (reset) menggunakan ikon reset.

Kedua, pemberian konteks pada halaman *drill-through*. Saat pengguna melakukan *drill-through* untuk melihat data yang lebih rinci, sangat penting bagi mereka untuk tidak kehilangan konteks. Solusinya adalah dengan memberikan subjudul yang dinamis atau jelas pada halaman *drill-through*, seperti halaman "*Drill-Through Total UIO*" atau "*Drill-Through Hide on FMS*". Ini secara instan memberi tahu pengguna data apa yang sedang mereka lihat.

Ketiga, penyediaan konteks data untuk validasi. Untuk membangun kepercayaan, *dashboard* harus transparan. Setiap halaman detail harus disertai dengan index atau hitungan jumlah baris (*count rows*). Ini adalah fitur sederhana namun krusial yang memungkinkan pengguna memvalidasi berapa banyak item data yang sedang ditampilkan atau terpengaruh oleh filter yang mereka terapkan.

Keempat adalah transparansi keterbaruan data. Visualisasi seperti "*Card Latest Data Sync*" sangat penting untuk dipastikan keterbaruannya dan visibilitasnya. Visual ini ditempatkan di lokasi yang mudah terlihat dan berfungsi untuk memberi tahu *user* kapan terakhir kali data *refresh*. Ini adalah hal utama dalam membangun kepercayaan dalam data, karena memastikan pengguna sadar bahwa mereka membuat keputusan berdasarkan data yang paling terbaru.