BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

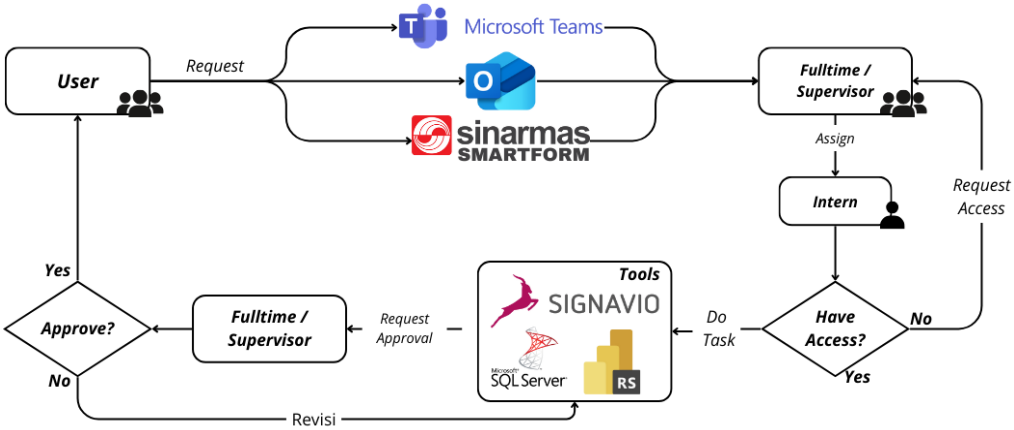
3.1.1 Kedudukan

SMSC berfungsi sebagai penyedia layanan pendukung terintegrasi bagi seluruh unit bisnis PT SMART Tbk, yang menangani berbagai fungsi non-operasional namun sangat krusial dalam menjaga efisiensi, efektivitas, dan kelancaran proses bisnis perusahaan. Divisi-divisi utama yang dikelola oleh SMSC mencakup Information Technology (IT), Human Resources (HR), Finance, General Affairs (GA), Human Resources Business Partner (HRBP), serta Center of Excellence (CoE). SMSC dibentuk untuk menyederhanakan dan mengintegrasikan proses administratif dan operasional, sehingga unit-unit bisnis seperti PT SMART Tbk dapat fokus pada core business-nya (misalnya produksi kelapa sawit dan produk turunannya), sementara tugas-tugas penunjang ditangani secara terpusat oleh SMSC.

Salah satu divisi strategis di dalam SMSC adalah Center of Excellence (CoE), yang memiliki tanggung jawab utama dalam mengembangkan serta mengimplementasikan inisiatif perbaikan dan transformasi proses bisnis. Divisi ini terdiri dari empat bagian utama, yaitu Operational Accelerator (OA), Operational Excellence (OE), Strategic Project, dan Learning and Development (LnD). Masing-masing bagian memiliki fokus khusus, mulai dari peningkatan efisiensi proses, perbaikan jangka panjang, pelaksanaan proyek-proyek strategis lintas divisi, hingga pengembangan kompetensi karyawan. Sebagai mahasiswa jurusan Sistem Informasi dengan peminatan Big Data, penempatan mahasiswa magang pada divisi Operational Accelerator (OA) sangat relevan. Hal ini dikarenakan OA berperan penting dalam pengolahan dan analisis data operasional yang digunakan untuk mendukung pengambilan keputusan berbasis data. Tugas-tugas seperti pembuatan dashboard, pengolahan data, dan analisis efisiensi sangat membutuhkan

kemampuan teknis seperti pemrograman, visualisasi data, dan pengelolaan database, yang sesuai dengan latar belakang akademis mahasiswa Sistem Informasi.

3.1.2 Koordinasi



Gambar 3.1 Alur Kerja Internship CoE

Alur kerja dan koordinasi selama menjalani program magang di PT SMART Tbk dalam divisi Center of Excellence (CoE), khususnya pada bagian Operational Accelerator (OA), berjalan secara dinamis dan berbasis proyek. Setiap pekerjaan dimulai dari adanya permintaan (request) yang diajukan oleh pengguna dari berbagai unit fungsional seperti Procurement, Finance, Human Resources (HR), General Affairs (GA), dan lainnya. Permintaan ini dapat disampaikan melalui beberapa saluran, seperti platform internal SMARTForm (website resmi untuk pembuatan tiket request), catatan di OneNote, atau melalui pesan langsung di Microsoft Teams kepada staf full-time atau supervisor di tim OA.

Setelah request diterima, supervisor atau staf akan mengevaluasi permintaan dan mendistribusikan pekerjaan kepada anggota tim, termasuk intern. Dalam beberapa kasus, intern bersama staf dan user akan melakukan pertemuan terlebih dahulu untuk mendiskusikan kebutuhan secara detail, menetapkan tujuan (goals), serta menentukan output yang diharapkan dari proyek tersebut. Setelah memahami kebutuhan, intern mulai mengerjakan tugas yang telah diberikan, yang dapat berupa pembuatan dashboard di Power BI, penulisan query SQL, atau perancangan alur bisnis menggunakan platform Signavio untuk kebutuhan Standard Work Instruction

(SWI). Selama proses pengerjaan, intern dapat berkoordinasi dengan staf atau user jika terdapat kendala atau jika dibutuhkan klarifikasi lebih lanjut.

Setelah tugas diselesaikan, hasil kerja akan dikirimkan kembali kepada staf full-time atau langsung kepada user untuk dilakukan review. Jika terdapat masukan atau revisi, intern perlu memperbaiki hasil kerja tersebut hingga sesuai dengan ekspektasi. Jika sudah dinyatakan sesuai, maka pekerjaan dianggap selesai. Pekerjaan yang dilakukan bersifat proyek, artinya tanggung jawab dapat berubah setiap beberapa minggu tergantung pada kebutuhan bisnis saat itu. Dalam satu periode, intern mungkin diminta untuk membuat dashboard baru dari awal (from scratch), sementara di periode lain dapat diberikan tugas untuk melakukan enhancement atau penyempurnaan pada dashboard yang sudah ada, atau memperbaiki error baik dari sisi tampilan maupun rumus DAX di Power BI.

3.2 Tugas dan Uraian Kerja Magang

Penempatan magang di divisi Operational Accelerator (OA), bagian dari Center of Excellence (CoE) PT SMART Tbk (Sinarmas Sentra Cipta), memberikan tanggung jawab utama dalam mendukung pengembangan dan pemeliharaan dashboard berbasis data. Tugas utama meliputi pembuatan dashboard baru sesuai kebutuhan berbagai divisi seperti Procurement, Finance, HR, dan General Affairs, serta melakukan maintenance, enhancement, debugging, dan fixing terhadap dashboard yang sudah berjalan. Proyek ini dimulai dari proses requirement gathering bersama user, dilanjutkan dengan desain, pengolahan data, dan visualisasi.

Selain pengembangan dashboard, peran intern juga mencakup dokumentasi teknis dan fungsional, pemetaan alur data, serta evaluasi efektivitas dashboard berdasarkan masukan user. Intern juga dilibatkan dalam pengumpulan dan analisis data dari berbagai sumber sistem (SAP, Excel, dll), validasi data untuk menjaga akurasi, serta menyusun rekomendasi perbaikan tampilan atau struktur informasi berdasarkan standar visualisasi perusahaan. Tugas yang dijalankan bersifat proyek dan dinamis, sehingga tanggung jawab dapat berganti setiap beberapa minggu tergantung kebutuhan perusahaan. Penugasan diberikan oleh staf atau supervisor OA, dan dikoordinasikan melalui meeting

berkala atau komunikasi langsung. Pada tabel 3.1 berikut adalah uraian dari pelaksanaan kerja magang:

Tabel 3.1 Uraian Tugas dan Waktu Pelaksanaan Magang

No	Deskripsi Kegiatan	Minggu ke-	Tanggal Mulai	Tanggal Selesai
1	Pengenalan lingkungan serta transfer knowledge mengenai sistem kerja di Sinarmas Sentra Cipta	1	6 Januari 2025	14 January 2025
Dashboard Development & Visualization				
2	Dashboard Development, Enhancement, and Maintenance	1 - 18	9 Januari 2025	5 Juni 2025
Database				
3	Pembuatan SQL untuk Pengambilan Email Vendor Terkait TAX/Finance (UNION ALL)	10-11	7 Maret 2025	11 Maret 2025

(Sumber olahan peneliti, 2025)

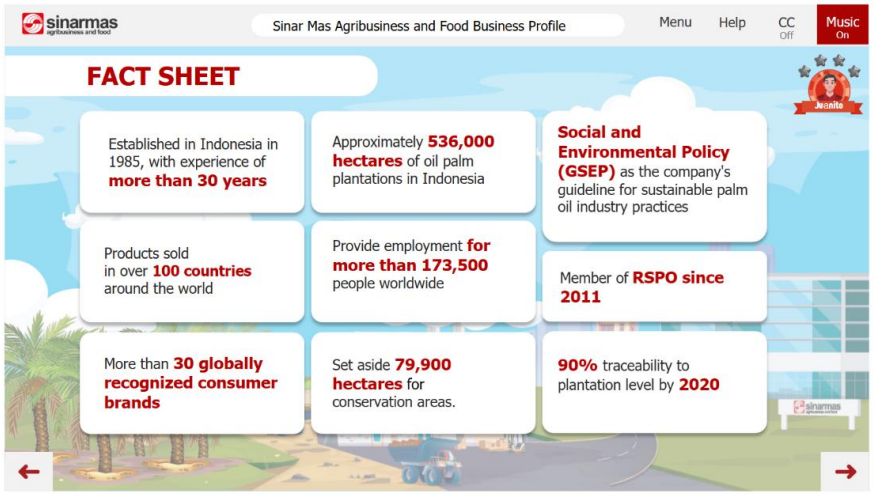
3.2.1 Pengenalan Lingkungan Kerja dan Flow Bisnis SinarMas serta Transfer Knowledge Mahasiswa Magang Sebelumnya

Pada minggu pertama pelaksanaan kegiatan magang, mahasiswa intern terlebih dahulu menjalani proses pengenalan lingkungan kerja untuk memahami dinamika serta budaya yang ada di PT SMART Tbk. Proses ini diawali dengan sesi perkenalan bersama seluruh anggota divisi Center of Excellence (CoE), di mana intern diperkenalkan kepada supervisor, staff, serta rekan-rekan kerja lain dalam satu divisi seperti yang terlihat pada Gambar 3.2. Dalam sesi ini, intern diperkenalkan kepada supervisor, staf, serta rekan-rekan kerja dalam satu tim untuk membangun komunikasi yang baik serta mengenali peran masing-masing individu dalam proses kerja.

Gambar 3.2 Pengenalan dengan anggota divisi CoE PT Smart Tbk

Setelah itu, intern mengikuti sesi pengenalan perusahaan secara umum yang membahas struktur organisasi PT SMART Tbk, nilai-nilai inti perusahaan, serta proses bisnis utama yang dijalankan oleh masing-masing divisi, termasuk sektor Upstream dan Downstream. Pada tahap ini, para intern diberikan materi pembelajaran dari platform internal PT SMART Tbk yang berisi video edukasi dan dilengkapi dengan kuis untuk menguji pemahaman terhadap materi yang telah disampaikan seperti yang diperlihatkan pada Gambar 3.3 dan Gambar 3.4. Karena akun intern belum tersedia pada hari pertama hingga hari kedua, proses pembelajaran ini dilakukan dengan menggunakan akun supervisor yang telah disediakan.

Gambar 3.3 Materi Pengenalan Perusahaan

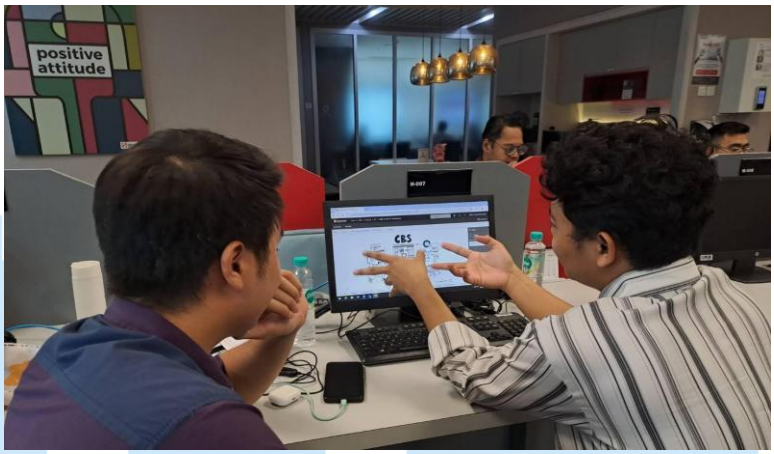


Gambar 3.4 Fact Sheet Sinarmas Agriculture and Food

Sesi pengenalan dan pembelajaran ini dilakukan secara individu, di mana masing-masing intern menggunakan laptop pribadi untuk mengakses materi secara mandiri, bukan dalam bentuk diskusi kelompok atau pembelajaran bersama. Materi yang disampaikan mencakup informasi mendalam mengenai sejarah dan struktur PT SMART Tbk, budaya kerja, serta visi dan misi yang menjadi landasan utama dalam menjalankan operasional bisnis. Selain itu, terdapat penjelasan khusus mengenai Center of Excellence (CoE) sebagai salah satu bagian penting dalam mendukung efisiensi dan efektivitas proses bisnis perusahaan. Intern juga dikenalkan dengan empat pilar utama dalam CoE, yaitu Operational Accelerator (OA), Operational Excellence (OE), Strategic Project, serta Learning and Development (LnD).

Setelah menyelesaikan sesi pengenalan perusahaan, pada hari keempat, intern melanjutkan kegiatan dengan mengikuti sesi transfer knowledge dari mahasiswa magang sebelumnya yang telah lebih dulu menjalani program magang selama satu tahun. Transfer knowledge ini dilakukan secara langsung oleh Dani Muhammad Lutfi, yang pada saat itu sedang berada di akhir masa magangnya dan akan menyelesaikan programnya pada bulan Februari. Dalam sesi ini, Dani membagikan penjelasan menyeluruh mengenai alur kerja yang biasa dijalankan, termasuk bagaimana proses permintaan pembuatan dashboard diterima, dikerjakan, dan dikomunikasikan kembali kepada tim terkait. Selain itu, Dani juga memberikan

pengajaran mengenai penggunaan Power BI secara efektif dan efisien, mulai dari pengolahan data, pembuatan visualisasi, hingga penulisan rumus DAX yang optimal untuk kebutuhan pelaporan dan analisis.

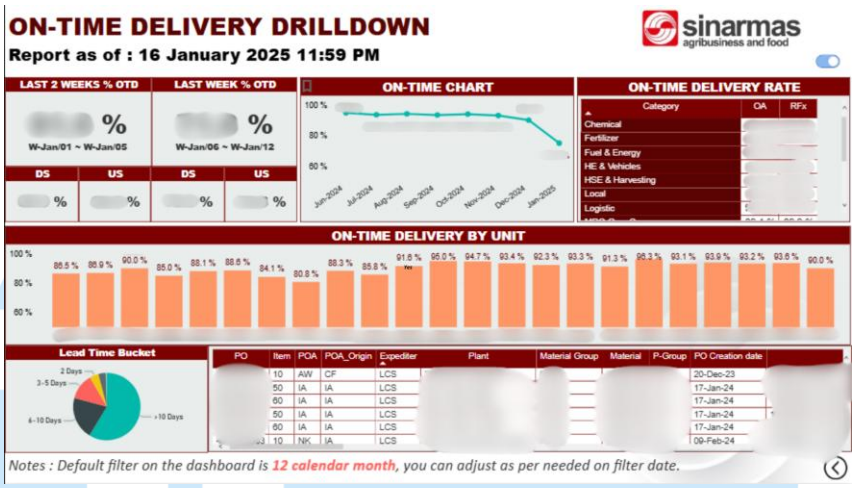


Gambar 3.5 Knowledge Transfer dengan Rekso

Proses transfer knowledge juga dilakukan oleh Rekso Jumantoro, salah satu intern yang memberikan penugasan lanjutan terkait dashboard *Citi Statement*. Dalam penjelasannya, Rekso menjabarkan tujuan pembuatan dashboard tersebut, penjelasan logika bisnis yang digunakan, struktur kolom dan metrik, serta sumber data yang menjadi acuan utama. Informasi ini diberikan agar intern dapat melanjutkan pengerjaan dashboard yang telah ada dengan pemahaman yang utuh dan akurat, serta memastikan kesinambungan kualitas dari pekerjaan yang telah dilakukan sebelumnya.

3.2.2 Dashboard Development, Enhancement, and Maintenance

3.2.2.1. Memperbaiki persentase dan line chart pada dashboard CBS-Proc-OnTime_Delivery



Gambar 3.6 Dashboard CBS-Proc-OnTime_Delivery

Tugas dashboard pertama yang diberikan kepada Intern adalah melakukan perbaikan pada dashboard *CBS-Proc-OnTime-Delivery*, yaitu dashboard yang dibuat oleh tim Central Business Services (sebutan untuk divisi CoE) untuk memantau kinerja pengiriman barang berdasarkan Purchase Order (PO) untuk procurement. Dashboard ini menampilkan berbagai metrik penting, seperti persentase On-Time Delivery (OTD) untuk dua minggu terakhir (*last 2 weeks*) dan minggu terakhir (*last week*), OTD rate selama 12 bulan terakhir, OTD rate berdasarkan kategori barang dan unit perusahaan, serta tabel detail data PO. Permasalahan pada dashboard ini diinformasikan oleh tim Procurement kepada Intern. Permasalahan pertama adalah visualisasi OTD rate selama 12 bulan terakhir yang hanya menampilkan 11 bulan. dan belum tersedianya indikator untuk persentase OTD dalam dua minggu terakhir.

Gambar 3.7 Request perbaikan dashboard CBS-Proc-OnTime_Delivery oleh user

Permasalahan pertama diselesaikan dengan melakukan penyesuaian pada pengaturan filter di kolom "monthname" yang sebelumnya belum sepenuhnya diaktifkan, sehingga grafik OTD rate hanya menampilkan 11 bulan terakhir dan tidak mencakup bulan terbaru. Intern mengaktifkan kembali seluruh nilai pada kolom "monthname" melalui tab filter pada visualisasi. Langkah ini memungkinkan tampilan data yang mencakup keseluruhan 12 bulan terakhir, sehingga visualisasi menjadi utuh dan dapat digunakan secara optimal untuk pemantauan kinerja pengiriman.

Permasalahan kedua berkaitan dengan belum tersedianya indikator persentase *On-Time Delivery* (OTD) untuk dua minggu terakhir (*last 2 weeks*), yang dibutuhkan oleh tim Procurement untuk memantau kinerja pengiriman dalam jangka waktu pendek. Untuk menjawab kebutuhan tersebut, Intern mengembangkan formula DAX baru dengan nama GRdt last2weeks yang bertujuan mengidentifikasi data *Goods Receipt* (GR) dalam dua minggu terakhir. Formula ini memanfaatkan beberapa variabel, yaitu last_gr_date sebagai tanggal penerimaan barang terakhir, today date sebagai referensi waktu saat ini, dan filter date untuk penyesuaian logika berdasarkan tahun. Fungsi WEEKNUM digunakan untuk menghitung nomor minggu dari tanggal GR dan tanggal hari ini. Formula ini juga dirancang untuk menangani kondisi khusus, seperti transisi tahun, yang dapat menyebabkan data dari dua minggu terakhir terbagi ke dalam dua tahun berbeda.

Berikut adalah kode DAX yang dikembangkan untuk menentukan minggu dari GR dalam dua minggu terakhir:

Name GRdt last2weeks | Whole number | Whole number | $ % | 0

```
GRdt last2weeks =
// IF(YEAR([last_gr_date]) = YEAR([filter date])
// && WEEKNUM([last_gr_date],2) >= WEEKNUM([today date],2)-2,0,1)

VAR _MAX_LAST_GR = MAX(PO_Details[last_gr_date])
VAR _WEEK_GR = WEEKNUM(PO_Details[last_gr_date],1)
VAR _WEEK_TODAY = WEEKNUM([today date],1)
VAR _LAST_WEEK_LAST_YEAR = WEEKNUM(CALCULATE(MAX([last_gr_date]),
    FILTER(ALL(PO_Details),YEAR(PO_Details[last_gr_date]) = YEAR(PO_Details[today date]) - 1)),1)
VAR _WEEK1 = IF(_WEEK_TODAY = 1,
    IF( YEAR([last_gr_date]) = YEAR([filter date])-1
    && _WEEK_GR in {_LAST_WEEK_LAST_YEAR,_LAST_WEEK_LAST_YEAR-1},1))
VAR _WEEK2 = IF(_WEEK_TODAY = 2,
    IF( (YEAR([last_gr_date]) = YEAR([filter date])-1 && _MAX_LAST_GR in {_LAST_WEEK_LAST_YEAR} )
        || (YEAR([last_gr_date]) = YEAR([filter date]) && _WEEK_TODAY-1 = 1)
        ,1))

RETURN
_WEEK_GR
```

Gambar 3.8 Kode Dax GRdt last2weeks

Selanjutnya, hasil perhitungan ini digunakan untuk memperoleh persentase OTD khusus dua minggu terakhir yang kemudian divisualisasikan pada dashboard, sehingga dapat memberikan informasi yang lebih akurat dan relevan bagi proses evaluasi performa pengiriman.Rumus DAX-nya adalah sebagai berikut:

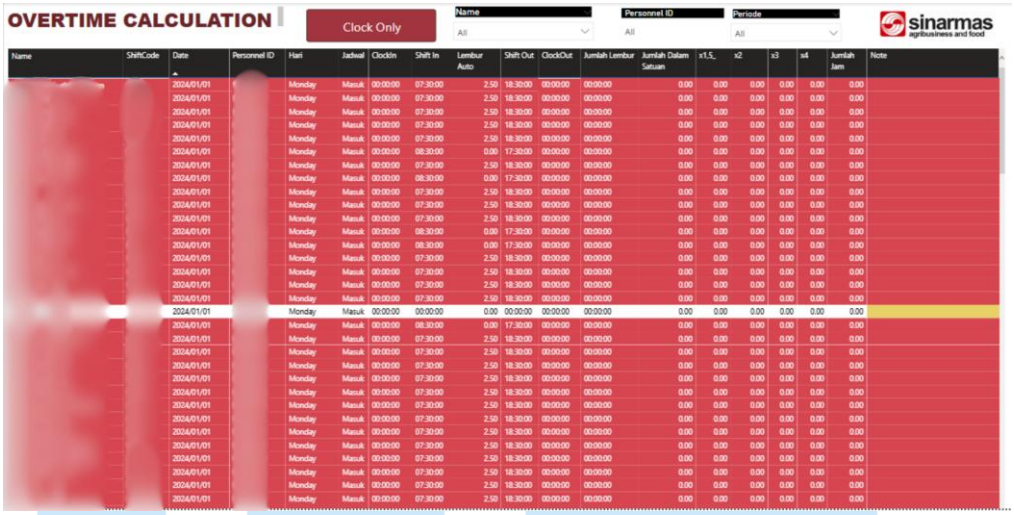
Name #OTD %Last2W... | Home table PO_Details | Format Percentage

```
#OTD %Last2Weeks = CALCULATE([#OTD %Rate], FILTER('PO_Details',[GRdt last2weeks]=1))+0
```

Gambar 3.9 Kode Dax #OTD %Last2Weeks

Measure ini menggunakan hasil dari GRdt last2weeks sebagai filter, kemudian menghitung kembali persentase OTD hanya untuk subset PO yang termasuk dalam dua minggu terakhir. Tambahan +0 digunakan untuk memastikan hasil tetap dalam format numerik bahkan ketika tidak ada data dalam periode tersebut.

3.2.2.2. Meningkatkan dashboard GA-Overtime-Calculation Time untuk highlight OB yang telat



Gambar 3.10 Dashboard GA-Overtime-Calculation Time

PT Sinar Mas Agribusiness and Food menggunakan sistem outsourcing untuk sebagian tenaga kerja operasional seperti petugas kebersihan (janitor), teknisi, dan staf keamanan. Karena status mereka bukan karyawan tetap, maka diperlukan sistem pemantauan khusus untuk mencatat dan menghitung jam kerja harian mereka secara akurat. Pemantauan ini bertujuan untuk mengetahui siapa saja tenaga kerja yang berhak menerima tambahan upah lembur, serta siapa yang mengalami pemotongan gaji akibat keterlambatan atau ketidaksesuaian jadwal. Oleh karena itu, kehadiran dan aktivitas kerja setiap tenaga kerja outsourcing harus dikaitkan secara langsung dengan jadwal kerja mereka (berdasarkan Shift Code), termasuk waktu masuk dan keluar yang tercatat melalui sistem absensi berbasis *face recognition*.

Sumber data untuk dashboard ini berasal dari file Excel manual yang diekstrak dari sistem pengenalan wajah. Pembaruan data dilakukan secara otomatis setiap 15 menit, dimulai dari pukul 08:00 hingga 18:00 setiap hari. Di halaman kalkulasi, terdapat komponen penting seperti kode shift, status kehadiran, jam masuk dan keluar, hingga perhitungan lembur otomatis dan manual. Kriteria lembur disusun dalam beberapa kategori (1.5, 2, 3, 4) sesuai durasi kerja tambahan dan ketepatan waktu kehadiran.

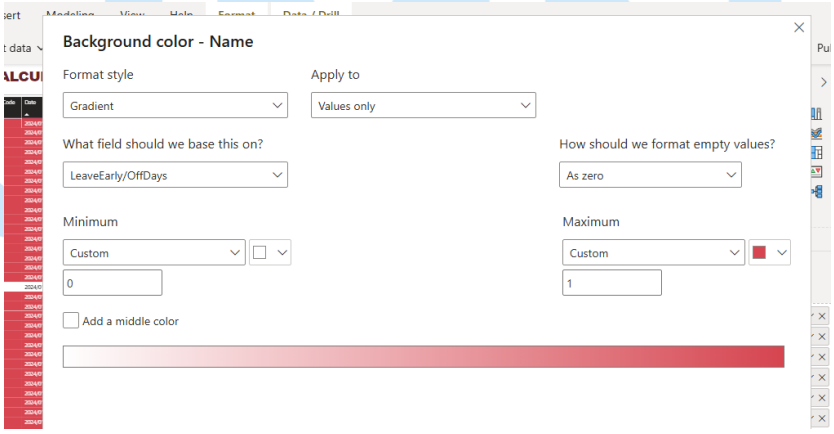
Pada pengembangan Dashboard GA-Overtime-Calculation, salah satu fitur

penting yang diimplementasikan adalah mekanisme penandaan otomatis (highlight) berwarna merah untuk menandai kondisi tertentu, seperti karyawan yang pulang lebih awal (Leave Early) atau sedang tidak masuk karena hari libur (Off Days). Fitur ini dibuat menggunakan rumus DAX dengan nama *LeaveEarly/OffDays*. Rumus ini mengevaluasi dua kondisi utama, yaitu jika waktu pulang aktual (Jam Keluar) lebih awal dari waktu pulang yang dijadwalkan (Shift Out), atau jika *Shift Code* memiliki nilai "Off". Jika salah satu kondisi terpenuhi, maka measure mengembalikan nilai 1, sedangkan jika tidak, nilainya 0. Berikut adalah rumus yang digunakan:

```
LeaveEarly/OffDays =
IF(
  TIMEVALUE('All Transactions'[Jam Keluar]) < TIMEVALUE('All Transactions'[Shift Out])
   ,1
     ,IF(SELECTEDVALUE('All Transactions'[ShiftCode]) = "Off"
            ,1
     ,0))
```

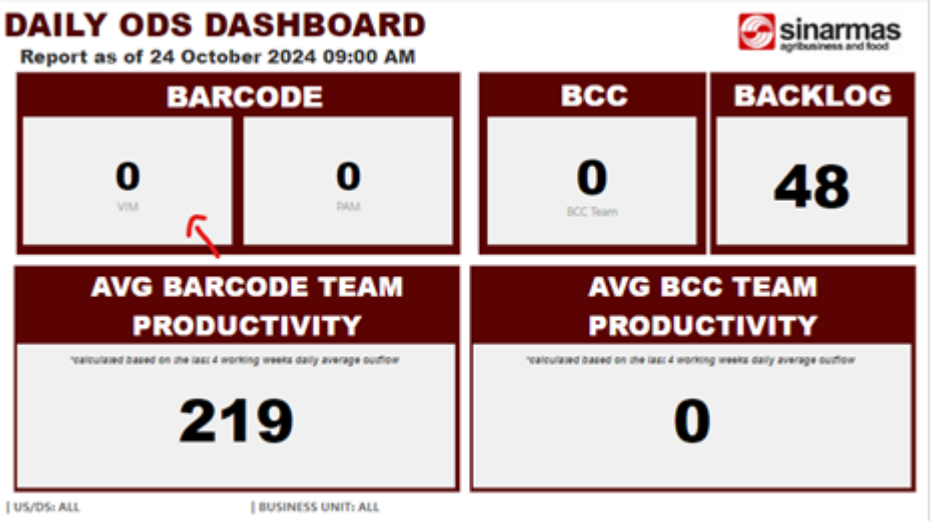
Gambar 3.11 Kode Dax LeaveEarly/OffDays

Measure ini kemudian digunakan dalam fitur *Conditional Formatting* pada Power BI, khususnya untuk pewarnaan latar belakang pada kolom seperti Remarks atau ClockOut. Jika nilai yang dikembalikan adalah 1, maka sel akan diberi warna latar merah, menandakan adanya ketidaksesuaian jadwal seperti pulang lebih awal atau status off. Fitur ini memungkinkan tim Facility & Office Service untuk memantau kondisi kehadiran dan lembur dengan lebih cepat dan efisien, terutama karena dashboard diperbarui secara otomatis setiap 15 menit selama jam kerja aktif (08.00–18.00).



Gambar 3.12 Conditional formatting LeaLeaveEarly/OffDays

3.2.2.3. Update dashboard I2P-BDC dan input new source data dari KUBE DB

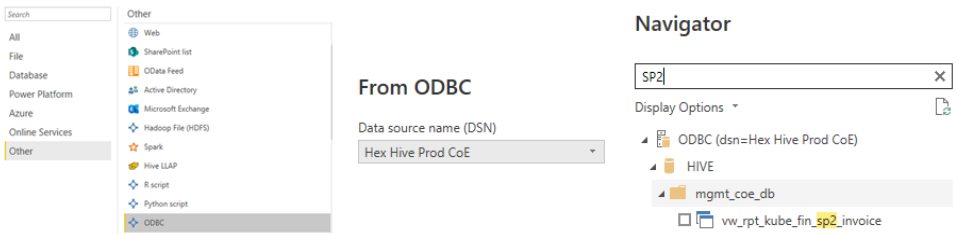


Gambar 3.13 Dashboard I2P-BDC

Dashboard FA-I2P-BDC Monitoring dikembangkan sebagai alat visualisasi data yang memberikan gambaran menyeluruh mengenai produktivitas harian tim CBS (Central Business Services), khususnya terkait volume pekerjaan dalam proses *One Day Service* (ODS). Dashboard ini dirancang untuk memantau performa dua tim utama, yaitu Barcode Team dan BCC Team, dengan fokus pada pengolahan data dari dua sumber utama: LDR SAP dan ODS Webtool. Pada awal tahun 2025, sumber data pada dashboard ini diperluas dengan penambahan data dari form Kube, yang dikumpulkan melalui platform internal dan ditampung di dalam database HEX Apache. Penambahan ini bertujuan untuk meningkatkan akurasi dan kelengkapan data dalam pelaporan volume kerja harian, serta memperkaya konteks analisis performa yang sebelumnya hanya mengandalkan data dari sistem SAP dan Webtool.

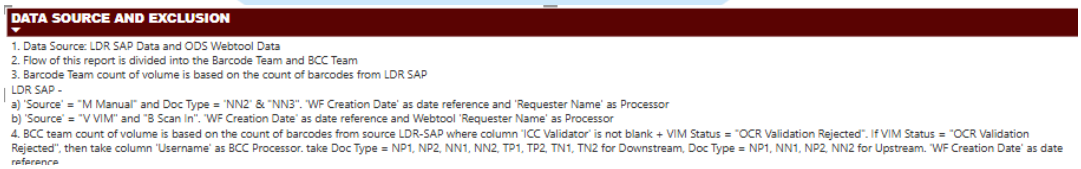
Untuk mengintegrasikan data dari form Kube ke dalam dashboard FA-I2P-BDC Monitoring, dilakukan proses pengambilan data (*data extraction*) menggunakan Power BI. Proses ini diawali dengan memilih opsi “Get Data”, kemudian memilih kategori “Other” dan selanjutnya memilih koneksi melalui ODBC (Open Database Connectivity). Dari daftar koneksi yang tersedia, dipilih sumber data Hex Hive Prod CoE, yaitu database milik divisi Center of Excellence

(CoE) yang menyimpan berbagai tabel terkait proses bisnis internal. Salah satu tabel yang dimanfaatkan adalah vw_rpt_kube_fin_sp2_invoice, yang berisi data invoice dari sistem form Kube. Setelah berhasil terhubung, tabel tersebut dimuat ke dalam Power BI dan dikaitkan dengan tabel-tabel lain yang relevan di dalam model data dashboard.



Gambar 3.14 Pengambilan data kube untuk I2P-BDC

Setelah proses integrasi data dari form Kube ke dalam dashboard FA-I2P-BDC Monitoring berhasil dilakukan, langkah selanjutnya adalah membangun *logic* perhitungan untuk menghitung volume kerja harian dari Barcode Team. Untuk tujuan ini, dikembangkan sebuah formula DAX dengan nama BarcodeTeam Count. Formula ini bertujuan untuk menghitung total jumlah barcode yang diproses, berdasarkan logika yang telah ditentukan oleh user dalam bagian *Data Source and Exclusion* pada dashboard.



Gambar 3.15 Rules logic dashboard I2P-BDC

Perhitungan dimulai dengan mendefinisikan variabel Man, yang menghitung jumlah barcode dari data LDR SAP dengan sumber "M Manual" dan dokumen bertipe *NN1*, *NN2*, *NN3*, serta tipe dokumen terkait lainnya yang telah ditentukan. Selanjutnya, variabel vb digunakan untuk menghitung barcode dengan sumber "V VIM" atau "B Scan In", yang juga berasal dari sistem LDR SAP. Sebagai bagian dari integrasi data form Kube, ditambahkan variabel FORMREF yang merepresentasikan jumlah unik formulir berdasarkan form_reference_no dari tabel KUBE, dengan kondisi seq_no = 1 sebagai penanda entri utama dari setiap formulir. Penggunaan DISTINCTCOUNT pada variabel ini memastikan bahwa setiap form

hanya dihitung satu kali meskipun terdapat duplikasi dalam pengisian data lanjutan. Akhirnya, formula DAX ini mengembalikan hasil penjumlahan dari ketiga variabel (Man, vb, dan FORMREF) untuk memberikan total kontribusi volume Barcode Team secara menyeluruh, mencakup baik data historis dari LDR SAP maupun data terbaru dari form Kube. Berikut adalah rumus yang digunakan:

```
BarcodeTeam Count =
VAR Man =
CALCULATE(COUNT('ConsolidatedDB'[Barcode]),
    FILTER('ConsolidatedDB','ConsolidatedDB'[Source] = "M Manual"),
    FILTER('ConsolidatedDB','ConsolidatedDB'[Document_Type] in
    {"NN1","NN2","NN3","NNS","NP1","NP2","NP3","TN1","TN2","TN3","TP1","TP2","TP3","TPA"}
    )
)

VAR vb =
CALCULATE(COUNT('ConsolidatedDB'[Barcode]),
    FILTER('ConsolidatedDB','ConsolidatedDB'[Source] = "V VIM" || 'ConsolidatedDB'[Source] = "B Scan In" )
)

VAR FORMREF =
    CALCULATE(
        DISTINCTCOUNT('KUBE'[form_reference_no]),
        'KUBE'[seq_no] = 1
    )

RETURN
Man + vb + 0
```

Gambar 3.16 Kode Dax BarcodeTeam Count

Setelah perhitungan volume total Barcode Team selesai dilakukan melalui *measure* BarcodeTeam Count, langkah berikutnya adalah menghitung nilai rata-rata harian pemrosesan barcode yang direpresentasikan oleh *measure* AveBarcodeFlow. Formula ini dirancang untuk menghitung rata-rata jumlah barcode yang diproses oleh tim per hari kerja selama periode waktu tertentu, sehingga dapat digunakan sebagai indikator produktivitas.

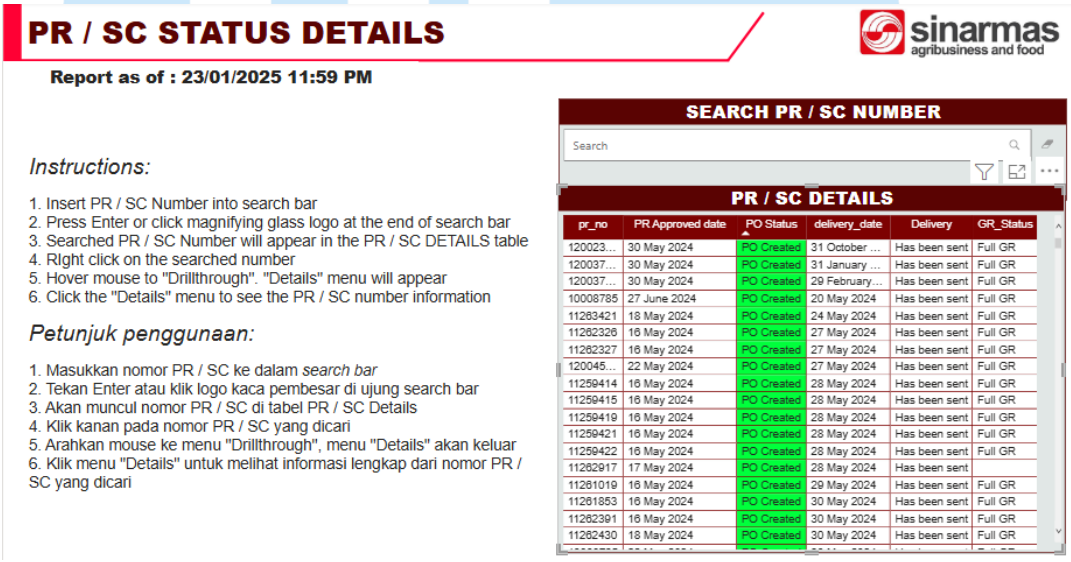
```
AveBarcodeFlow = (CALCULATE(
    INT(
        AVERAGEX(KEEPFILTERS(VALUES('Calendar'[Date])),
            CALCULATE([BarcodeTeam Count])))))
*COUNT(Calendar[Date])
/(IF(SUM(Calendar[Workday]) = 0, 1, SUM(Calendar[Workday])))
```

Gambar 3.17 Kode Dax AveBarcodeFlow

Perhitungan dimulai dengan menggunakan fungsi AVERAGEX terhadap nilai unik tanggal (Calendar[Date]) yang difilter sesuai konteks visualisasi. Setiap tanggal dihitung nilai BarcodeTeam Count-nya, kemudian dirata-ratakan secara

keseluruhan. Hasil rata-rata tersebut kemudian dibulatkan menjadi bilangan bulat dengan fungsi INT() untuk mempermudah interpretasi data. Selanjutnya, nilai rata-rata tersebut dikalikan dengan jumlah hari dalam konteks kalender aktif (COUNT(Calendar[Date])) untuk mendapatkan total aliran volume. Langkah terakhir adalah membagi hasil tersebut dengan jumlah hari kerja (SUM(Calendar[Workday])) guna menyesuaikan perhitungan hanya terhadap hari kerja saja. Untuk menghindari pembagian dengan nol, digunakan fungsi IF() sebagai kontrol logika, yaitu apabila jumlah hari kerja nol maka digantikan dengan nilai satu sebagai penyebut.

3.2.2.4. Memperbaiki dashboard PR & PO dan update data



Gambar 3.18 Dashboard PR & PO

Dalam proses pengadaan barang dan jasa, keberadaan sistem pemantauan yang efisien dan terintegrasi menjadi sangat krusial untuk memastikan setiap tahapan berjalan sesuai dengan waktu dan prosedur yang telah ditentukan. Proses pengadaan sering kali melibatkan banyak dokumen seperti Purchase Requisition (PR) dan Shopping Cart (SC), yang harus ditelusuri secara tepat untuk menghindari keterlambatan pengiriman, kesalahan data, atau ketidaksesuaian antara permintaan dan penerimaan barang. Dashboard PR/SC Monitoring berfungsi sebagai alat visualisasi interaktif yang mempermudah proses pemantauan dan penelusuran dokumen pengadaan seperti Purchase Requisition (PR) dan Shopping Cart (SC).

Dashboard ini dirancang agar pengguna dapat mencari nomor PR atau SC tertentu melalui kolom pencarian, kemudian melihat detail informasi terkait dokumen tersebut. Setelah memasukkan nomor PR atau SC dan menekan enter atau ikon pencarian, informasi yang relevan akan ditampilkan dalam tabel, mencakup kolom-kolom penting seperti nomor PR, tanggal persetujuan PR, status pembuatan Purchase Order (PO), tanggal estimasi pengiriman, status pengiriman, serta status penerimaan barang (Goods Receipt).

Pada dashboard PR/SC Monitoring ini, intern diberikan tugas untuk mengembangkan logik agar bisa menampilkan informasi ketika pengguna melakukan pencarian berdasarkan nomor Purchase Requisition (PR) atau Shopping Cart (SC). Logika baru ini dirancang untuk secara otomatis menampilkan tiga informasi penting terkait Purchase Order (PO), yaitu: status PO (apakah PO sudah dibuat atau belum), tanggal estimasi pengiriman (delivery date) berdasarkan data yang tercatat dalam sistem, serta status pengiriman (apakah barang sudah dikirim atau belum).Untuk menampilkan status apakah Purchase Order (PO) telah dibuat dan apakah barang sudah dikirim, berikut adalah kodenya yang dapat dilihat pada gambar 3.24 dan 3.25

```
PO Created =
IF(
    ISBLANK('PR & PO Details'[po_creation_date]),
    "PO Not Created",
    "PO Created"
)
```

Gambar 3.19 Kode Dax PO Created

Kode di atas berfungsi untuk memeriksa apakah tanggal pembuatan PO (po_creation_date) tersedia. Jika kolom tersebut kosong (ISBLANK), maka sistem akan menampilkan status "PO Not Created". Sebaliknya, jika tanggal tersedia, maka status yang ditampilkan adalah "PO Created".

```
Delivery =
IF(
    ISBLANK('PR & PO Details'[delivery_date]),
    "Have not been sent",
    "Has been sent"
)
```

Gambar 3.20 Kode Dax Delivery

Sementara itu, kode untuk status pengiriman (Delivery) digunakan untuk mengevaluasi apakah tanggal pengiriman (delivery_date) telah tercatat. Jika belum ada tanggal yang tercantum, maka sistem menampilkan status "Have not been sent". Jika tanggal tersedia, maka artinya barang sudah dikirim dan status yang ditampilkan adalah "Has been sent". Logika ini membantu pengguna dashboard dalam memantau proses pengadaan secara efisien, mulai dari pembuatan PO hingga pengiriman barang.

pr_no	PR Approved date	PO Status	delivery_date	Delivery	GR_Status
	30 May 2024	PO Created	31 October ...	Has been sent	Full GR
	30 May 2024	PO Created	31 January ...	Has been sent	Full GR
	30 May 2024	PO Created	29 February...	Has been sent	Full GR
	27 June 2024	PO Created	20 May 2024	Has been sent	Full GR
	18 May 2024	PO Created	24 May 2024	Has been sent	Full GR
	16 May 2024	PO Created	27 May 2024	Has been sent	Full GR
	16 May 2024	PO Created	27 May 2024	Has been sent	Full GR
	22 May 2024	PO Created	27 May 2024	Has been sent	Full GR
	16 May 2024	PO Created	28 May 2024	Has been sent	Full GR
	16 May 2024	PO Created	28 May 2024	Has been sent	Full GR
	16 May 2024	PO Created	28 May 2024	Has been sent	Full GR
	16 May 2024	PO Created	28 May 2024	Has been sent	Full GR
	16 May 2024	PO Created	28 May 2024	Has been sent	Full GR
	17 May 2024	PO Created	28 May 2024	Has been sent	
	16 May 2024	PO Created	29 May 2024	Has been sent	Full GR
	16 May 2024	PO Created	30 May 2024	Has been sent	Full GR
	16 May 2024	PO Created	30 May 2024	Has been sent	Full GR
	18 May 2024	PO Created	30 May 2024	Has been sent	Full GR

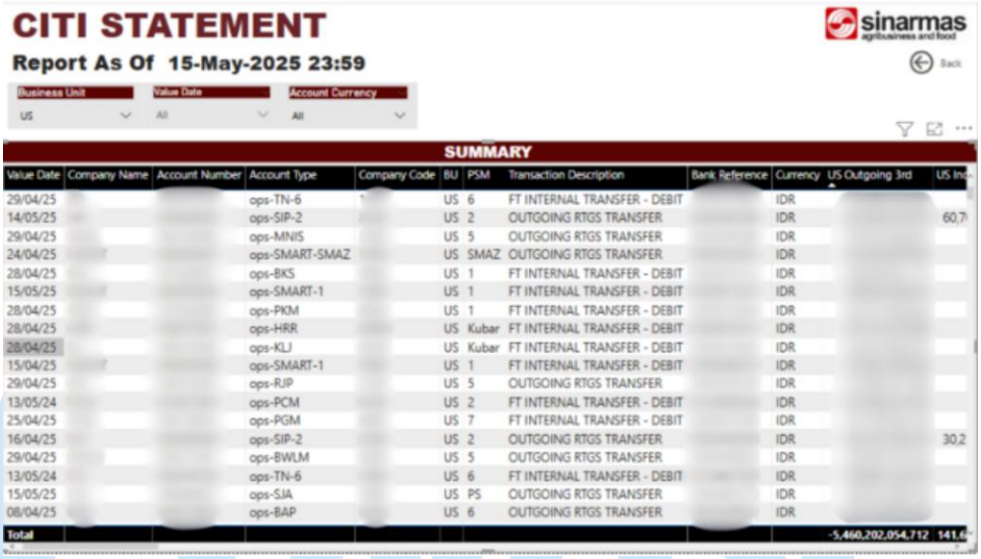
Gambar 3.21 Hasil dashboard PR & PO

Melalui kode PO Created, sistem akan menampilkan informasi apakah PO dari PR/SC tersebut sudah dibuat atau belum, sedangkan melalui kode Delivery, pengguna dapat melihat apakah barang dari PO tersebut sudah dikirim atau masih dalam proses seperti yang dapat dilihat pada gambar 3.26 diatas. Dengan adanya logika ini, dashboard menjadi lebih informatif dan responsif, membantu pengguna dalam melakukan monitoring proses pengadaan secara efisien tanpa perlu

membuka sistem lain untuk memverifikasi status dokumen. Kontribusi intern ini secara langsung meningkatkan kegunaan dan kelengkapan dashboard

3.2.2.5. Pemeliharaan dashboard Citi Statement (rekap rekening, formula baru, company code)

Dalam proses transaksi bisnis, Sinarmas menggunakan Citibank sebagai bank utama untuk mengelola berbagai aktivitas keuangan lintas unit dan entitas. Citibank dipilih karena mampu menyediakan layanan perbankan korporasi yang terintegrasi dan sesuai dengan kebutuhan bisnis berskala besar seperti Sinarmas. Sejalan dengan itu, dashboard Citi Statement dikembangkan sebagai alat pemantauan transaksi dan mutasi rekening Citibank yang digunakan oleh seluruh entitas di bawah naungan Sinarmas. Hal ini penting mengingat hampir seluruh aktivitas keuangan perusahaan terpusat melalui rekening Citibank, sehingga diperlukan sistem pelaporan yang real-time untuk menjaga transparansi, akuntabilitas, dan efisiensi pengelolaan keuangan.



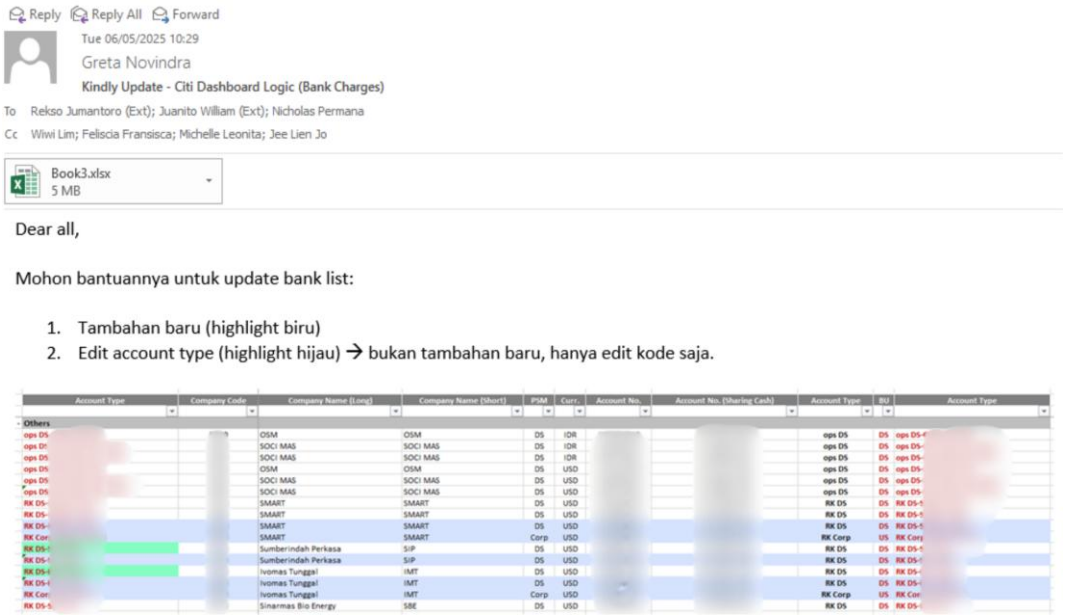
Value Date	Company Name	Account Number	Account Type	Company Code	BU	PSM	Transaction Description	Bank Reference	Currency	US Outgoing 3rd	US Inc
29/04/25			ops-TN-6		US	6	FT INTERNAL TRANSFER - DEBIT		IDR		
14/05/25			ops-SIP-2		US	2	OUTGOING RTGS TRANSFER		IDR		60,7
29/04/25			ops-MNIS		US	5	OUTGOING RTGS TRANSFER		IDR		
24/04/25			ops-SMART-SMAZ		US	SMAZ	OUTGOING RTGS TRANSFER		IDR		
28/04/25			ops-BKS		US	1	FT INTERNAL TRANSFER - DEBIT		IDR		
15/05/25			ops-SMART-1		US	1	FT INTERNAL TRANSFER - DEBIT		IDR		
28/04/25			ops-PKM		US	1	FT INTERNAL TRANSFER - DEBIT		IDR		
28/04/25			ops-HRR		US	Kubar	FT INTERNAL TRANSFER - DEBIT		IDR		
28/04/25			ops-KLJ		US	Kubar	FT INTERNAL TRANSFER - DEBIT		IDR		
15/04/25			ops-SMART-1		US	1	FT INTERNAL TRANSFER - DEBIT		IDR		
29/04/25			ops-RJP		US	5	OUTGOING RTGS TRANSFER		IDR		
13/05/24			ops-PCM		US	2	FT INTERNAL TRANSFER - DEBIT		IDR		
25/04/25			ops-PGM		US	7	FT INTERNAL TRANSFER - DEBIT		IDR		
16/04/25			ops-SIP-2		US	2	OUTGOING RTGS TRANSFER		IDR		30,2
29/04/25			ops-BWLM		US	5	OUTGOING RTGS TRANSFER		IDR		
13/05/24			ops-TN-6		US	6	FT INTERNAL TRANSFER - DEBIT		IDR		
15/05/25			ops-SJA		US	PS	OUTGOING RTGS TRANSFER		IDR		
08/04/25			ops-BAP		US	6	OUTGOING RTGS TRANSFER		IDR		
Total										-5,460,202,054,712	141,6

Gambar 3.22 Dashboard CITI Statement

Dashboard ini mencakup seluruh jenis transaksi dari tiga kategori rekening utama, yaitu Operation, Collection, dan Corporate. Rekening *Operation* berfungsi untuk meminta dana dari pusat guna mendukung operasional harian, seperti pembelian kebutuhan atau pembayaran vendor. Rekening *Collection* digunakan untuk menyetor kembali dana ke corporate, misalnya dari hasil penjualan atau

pendapatan lainnya. Sementara itu, rekening *Corporate* bertindak sebagai pusat dari seluruh aktivitas keuangan, baik sebagai sumber dana maupun penerima aliran dana dari seluruh unit bisnis.

Dalam proyek dashboard Citi Statement, intern bertugas untuk melakukan pemeliharaan dan pembaruan data berdasarkan arahan dari tim Central Finance (CF). Tugas utama intern meliputi tiga aspek penting: update rekap rekening, pembaharuan formula, serta penyesuaian company code. Setiap pembaruan dilakukan sebagai respons terhadap permintaan dari CF, misalnya ketika terdapat penambahan data baru seperti nomor rekening atau informasi company pemegang rekening. Dalam situasi tersebut, intern akan menginput data baru yang disediakan dalam bentuk file Excel ke dalam sumber data dashboard. Pada gambar 3.28 adalah contoh request dari CF terkait penambahan data baru:



Gambar 3.23 Request user Central Finance untuk penambahan data baru

Selain itu, jika terjadi perbedaan atau selisih data antara catatan keuangan CF dan data yang ditampilkan dalam dashboard, intern bertanggung jawab untuk memperbarui logic atau formula dashboard agar sesuai dengan referensi Excel dari CF seperti pada contoh request CF pada gambar 3.29 beserta dengan contoh logic pada excel dan pada DAX PowerBI nya pada gambar 3.30 dan 3.31.

Reply Reply All Forward

Thu 08/05/2025 14:03

Greta Novindra

Kindly Crosscheck - Citi Dashboard Difference (VD Apr 29, 2025)

To Nicholas Permana; Rekso Jumantoro (Ext); Juanito William (Ext)

Cc Wiwi Lim; Feliscia Fransisca; Michelle Leonita; Jee Lien Jo

You replied to this message on 13/05/2025 12:01.

Kindly Update - Citi Dashboard Logic (Bank Charges)
Outlook item

1.3. Citi Statement Automation - Dashboard Check (290425).xlsx
1 MB

Dear all,

Mohon bantuannya untuk crosscheck selisih pada file terlampir (sheet Comparison US dan sheet Comparison DS).

- Citi per CF → refer to sheet Citi (atau refer to file pada email terlampir)
- Citi per Dashboard → refer to sheet Dashboard-Details

Thank you.

Gambar 3.24 Request user Central Finance terkait selisih value data

```
=IF(JJ9="y";0;IF(T9>0;0;IF(IJ9<0;0;IF(JR9="Y";0;IF(JH9=0;IF(JE9>0;0;IF(AND(LEFT(H9;3)="ops";T9>0;BB9="INTRA DEPT TCL DEPT RP");T9;IF(AND(OR(RIGHT(H9;3)="KLJ";RIGHT(H9;3)=
"HRR");V9="OUTGOING RTGS TRANSFER";X9="RTGS");0;IF(LEFT(H9;3)="ops";IF(AND(T9<0;(OR(V9="GIRO CHARGE";V9="GIRO PAYMENT";V9="RTGS CHARGE";V9="OUTGOING
RTGS TRANSFER";V9="X-BORDER WIRE FT CHARGE";V9="FT OUTGOING - SWIFT")));T9;IF(AND(T9<0;JD9>0);T9;0));0))));0)))))
```

Gambar 3.25 Logic Excel US Outgoing 3rd

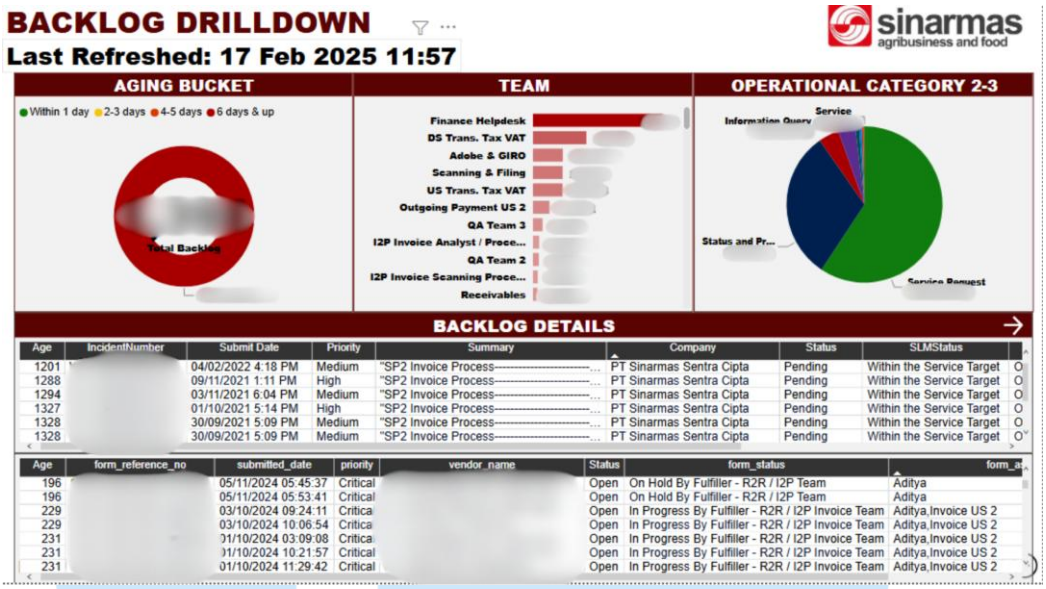
```
US Outgoing 3rd =

IF(
    'Citi Statement old'[FX Related] = "y",
    0,
    IF(
        'Citi Statement old'[Transaction Amount] > 0,
        0,
        IF(
            'Citi Statement old'[DS/US Incoming DME 2nd (Swap Buy-Out IDR)] < 0,
            0,
            IF(
                'Citi Statement old'[SBE - Special Rate] = "Y",
                0,
                IF(
                    'Citi Statement old'[US Pindah Dana Kubar] = 0,
                    IF(
                        'Citi Statement old'[US_Interco] > "0",
                        0,
                        IF(
                            LEFT('Citi Statement old'[Account Type (To Use)], 3) = "ops" &&
                            'Citi Statement old'[Transaction Amount] > 0 &&
                            'Citi Statement old'[Name/Address_16] = "INTRA DEPT TCL DEPT RP",
                            'Citi Statement old'[Transaction Amount],
                            IF(
                                (RIGHT('Citi Statement old'[Account Type (To Use)], 3) = "KLJ" ||
                                RIGHT('Citi Statement old'[Account Type (To Use)], 3) = "HRR") &&
                                'Citi Statement old'[Transaction Description] = "OUTGOING RTGS TRANSFER" &&
                                'Citi Statement old'[Payment Details] = "RTGS",
                                0,
                                IF(
                                    LEFT('Citi Statement old'[Account Type (To Use)], 3) = "ops",
                                    IF(
                                        'Citi Statement old'[Transaction Amount] < 0 &&
                                        (
                                            'Citi Statement old'[Transaction Description] = "GIRO CHARGE" ||
                                            'Citi Statement old'[Transaction Description] = "GIRO PAYMENT" ||
                                            'Citi Statement old'[Transaction Description] = "RTGS CHARGE" ||
                                            'Citi Statement old'[Transaction Description] = "OUTGOING RTGS TRANSFER" ||
                                            'Citi Statement old'[Transaction Description] = "X-BORDER WIRE FT CHARGE" ||
                                            'Citi Statement old'[Transaction Description] = "FT OUTGOING - SWIFT"
                                        ),
                                        'Citi Statement old'[Transaction Amount],
                                        IF('Citi Statement old'[Transaction Amount] < 0 && 'Citi Statement old'[US_Not_Interco] > "0",
                                            'Citi Statement old'[Transaction Amount],
                                            0
                                        )
```

Gambar 3.26 Logic DAX US Outgoing 3rd

Setelah Intern melakukan perbaikan logic, intern melakukan validasi untuk memastikan bahwa hasil yang ditampilkan sudah akurat dan selaras dengan catatan keuangan pusat. Peran ini sangat krusial untuk menjaga integritas dan konsistensi data keuangan, serta memastikan bahwa dashboard tetap menjadi sumber informasi yang faktual.

3.2.2.6. Update data dashboard Finance Helpdesk dan update formula pada visual dashboard



Gambar 3.27 Dashboard Finance Helpdesk

Serupa dengan pendekatan yang diterapkan pada dashboard I2P-BDC, pada Finance Helpdesk Dashboard ini intern juga melakukan integrasi data dari KUBE Database. Intern menggabungkan data dari dashboard utama dengan data tambahan yang berasal dari form KUBE, yang disimpan di dalam database HEX milik divisi Center of Excellence (CoE). Proses integrasi ini dilakukan melalui koneksi ODBC pada Power BI, di mana intern memilih koneksi ke sumber data *Hex Hive Prod CoE*

```
CountID2 =
VAR Base=
CALCULATE(DISTINCTCOUNT(ConsolidatedDB[WO/IncidentNumber]))+0

VAR FORMREF =
CALCULATE(DISTINCTCOUNT('Kube'[form_reference_no]))+0

RETURN
Base + FORMREF +  0
```

Gambar 3.28 Kode DAX CountID2

DAX CountID2 digunakan untuk menghitung total jumlah unik identifikasi pekerjaan atau dokumen yang berasal dari dua sumber data berbeda yang digabungkan dalam dashboard. Pertama, menghitung jumlah unik WO/IncidentNumber dari tabel ConsolidatedDB yang merepresentasikan data

Work Order atau insiden dari sistem utama. Kedua, menghitung jumlah unik form_reference_no dari tabel Kube, yaitu data tambahan yang berasal dari form KUBE yang disimpan di database HEX. Dengan menjumlahkan kedua nilai tersebut, DAX ini memberikan gambaran total transaksi atau pekerjaan yang tercatat dari kedua sumber data tersebut secara keseluruhan. Setelah DAX ini dibuat, intern kemudian melakukan pembaruan pada seluruh visualisasi dalam dashboard agar data dari KUBE dapat tampil dengan baik dan memberikan gambaran yang komprehensif sesuai dengan data terbaru yang telah diintegrasikan.

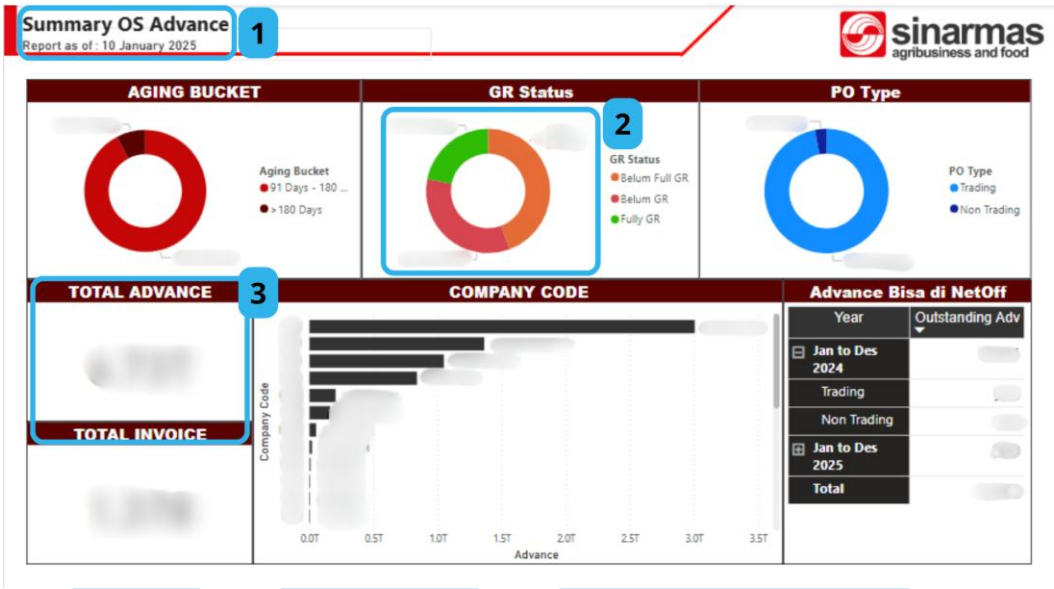
Age	form_reference_no	submitted_date	priority	vendor_name	Status	form_status	form_a
196	[redacted]	05/11/2024 05:45:37	Critical	[redacted]	Open	On Hold By Fulfiller - R2R / I2P Team	Aditya
196	[redacted]	05/11/2024 05:53:41	Critical	[redacted]	Open	On Hold By Fulfiller - R2R / I2P Team	Aditya
229	[redacted]	03/10/2024 09:24:11	Critica	[redacted]	Open	In Progress By Fulfiller - R2R / I2P Invoice Team	Aditya,Invoice US 2
229	[redacted]	03/10/2024 10:06:54	Critica	[redacted]	Open	In Progress By Fulfiller - R2R / I2P Invoice Team	Aditya,Invoice US 2
231	[redacted]	01/10/2024 03:09:08	Critica	[redacted]	Open	In Progress By Fulfiller - R2R / I2P Invoice Team	Aditya,Invoice US 2
231	[redacted]	01/10/2024 10:21:57	Critical	[redacted]	Open	In Progress By Fulfiller - R2R / I2P Invoice Team	Aditya,Invoice US 2
231	[redacted]	01/10/2024 11:29:42	Critical	[redacted]	Open	In Progress By Fulfiller - R2R / I2P Invoice Team	Aditya,Invoice US 2

Gambar 3.29 Tabel baru data KUBE untuk dashboard Finance Helpdesk

Selain itu, intern juga menambahkan tabel baru ke dalam dashboard Finance Helpdesk yang memanfaatkan data dari database HEX. Tabel baru ini berisi beberapa kolom penting seperti Age, form_reference_no, submitted_date, priority, vendor_name, Status, dan form_assignee. Penambahan tabel ini bertujuan untuk memperkaya informasi yang tersedia dalam dashboard sehingga pengguna dapat memperoleh insight yang lebih mendalam terkait pekerjaan dan proses yang sedang berjalan, khususnya yang berkaitan dengan form KUBE.

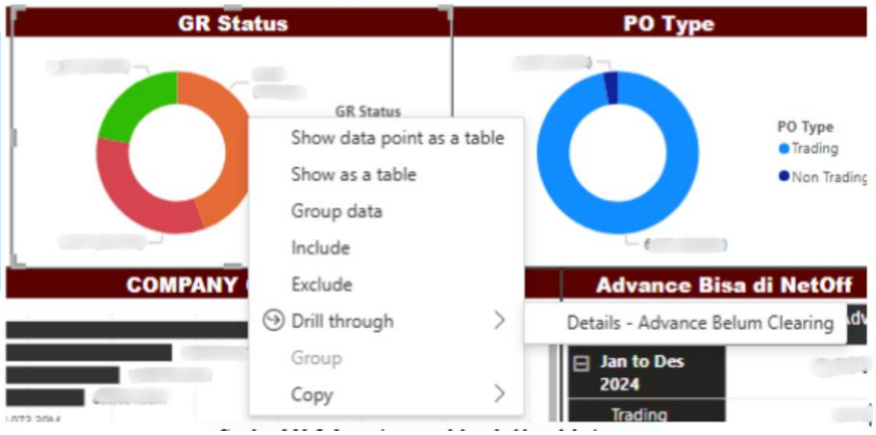
3.2.2.7. Memperbaiki visualisasi pada dashboard Vendor Advance Realization

Pada dashboard *Vendor Advance Realization*, intern diberikan tanggung jawab untuk melakukan sejumlah pengembangan dan penyempurnaan visualisasi guna meningkatkan akurasi serta kemudahan pemantauan proses realisasi uang muka (advance) vendor. Dashboard ini dirancang untuk memberikan pemahaman menyeluruh terhadap status advance yang belum dilakukan clearing (penyelesaian), dengan mempertimbangkan kebutuhan pengguna dan divisi terkait.



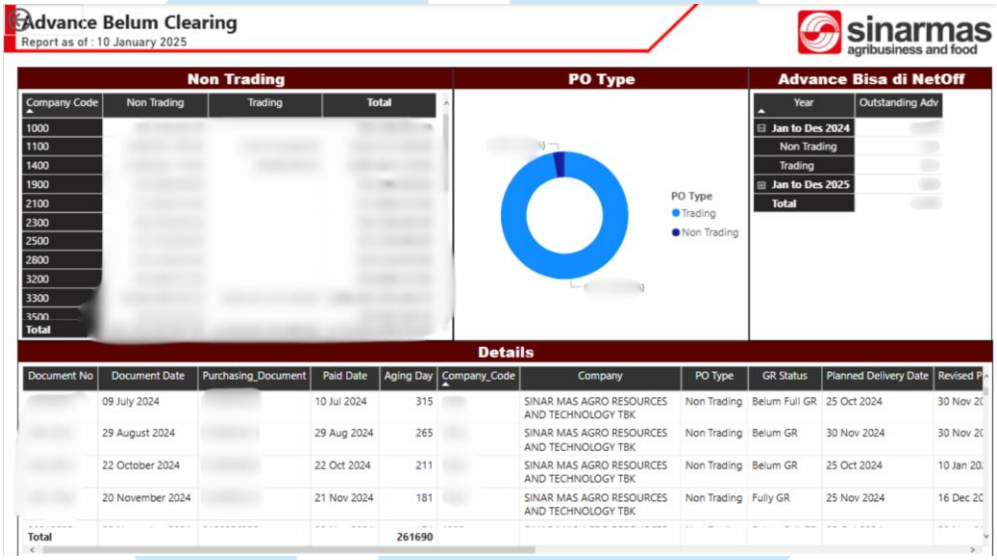
Gambar 3.30 Dashboard Vendor Advance Realization

Tugas pertama yang dilaksanakan intern adalah melakukan penyesuaian pada halaman *Advance Belum Clearing*. Penyesuaian ini dapat dilihat pada gambar3.35yang ditandai dengan nomor 1 sampai 3 yang mencakup penggantian judul pada bagian *Summary OS Advance* agar lebih representatif, serta melakukan modifikasi pada pewarnaan status *Goods Receipt (GR)* agar informasi yang ditampilkan dapat terbaca dengan lebih jelas dan menarik secara visual. Intern juga menambahkan tabel baru yang menampilkan total nilai advance yang belum terselesaikan, sehingga mempermudah pengguna dalam melihat ringkasan outstanding advance secara langsung.



Gambar 3.31 Fitur drillthrough ke halaman Detail Advance.

Selain itu, intern juga mengaktifkan fitur *drillthrough* pada seluruh visualisasi dalam dashboard guna meningkatkan interaktivitas dan kedalaman analisis data bagi pengguna. Fitur *drillthrough* memungkinkan pengguna untuk melakukan klik kanan pada elemen visual tertentu, seperti bar chart, pie chart, atau tabel, kemudian memilih opsi "Drillthrough" untuk berpindah secara langsung ke halaman *Detail Advance*.



Gambar 3.32 Halaman Detail Advance dashboard Vendor Advance Realization

Pada halaman Detail Advance ini, intern membangun tabel dengan kolom-kolom penting seperti *Total Advance GR* (yang dinamai ulang menjadi *Advance Bisa di NetOff*), *OS GR (Qty)*, *Document Date*, *Delivery Date*, *Incoterms*, dan *Due Date Payment*, sehingga dapat memberikan informasi yang lebih komprehensif

```
PO Qty =
VAR _PO_DP = LOOKUPVALUE(PO_DP[qty],PO_DP[purchasing_document],Invoice_DP[PO New])
VAR _PO_DP_Project = LOOKUPVALUE(PO_DP_Project[qty],PO_DP_Project[purchasing_document],Invoice_DP[PO New])
VAR _PO_Trading = LOOKUPVALUE(PO_Trading_DS[order_qty],PO_Trading_DS[purch_doc],Invoice_DP[PO New])

RETURN

IF(Invoice_DP[PO Type] = "PO Trading",_PO_Trading,
IF(_PO_DP <> BLANK(), _PO_DP, _PO_DP_Project))
```

Gambar 3.33 Kode DAX PO Qty

Sebagai bagian dari pengembangan dashboard *Vendor Advance Realization*, intern juga bertanggung jawab untuk membuat sejumlah kode DAX yang berfungsi untuk menghitung jumlah kuantitas pesanan (*Purchase Order Quantity*) dan kuantitas penerimaan barang (*Goods Receipt Quantity*) berdasarkan tipe dokumen

pembelian yang berbeda. Intern membuat kalkulasi DAX bernama PO Qty untuk mengidentifikasi jumlah kuantitas pesanan berdasarkan nilai yang di-*lookup* dari tiga tabel berbeda, yaitu PO_DP, PO_DP_Project, dan PO_Trading_DS. Fungsi LOOKUPVALUE digunakan untuk mencocokkan kolom *purchasing_document* atau *purch_doc* dengan nilai dari kolom Invoice_DP[PO New], yang kemudian dikondisikan berdasarkan tipe PO seperti "PO Trading".

```
GR Qty =
// CALCULATE(ABS(SUM(FAGLL03_GRIR_No_Clearing[Amount_in_doc_curr])),
//     FILTER(ALL(FAGLL03_GRIR_No_Clearing),FAGLL03_GRIR_No_Clearing[Purchasing_Doc] = Invoice_DP[Purchasir

VAR _GR = LOOKUPVALUE(PO_DP[GR qty],PO_DP[purchasing_document],Invoice_DP[PO New])
VAR _GR_Project = LOOKUPVALUE(PO_DP_Project[GR qty],PO_DP_Project[purchasing_document],Invoice_DP[PO New])
VAR _GR_Trading = LOOKUPVALUE(PO_Trading_DS[gr_quantity],PO_Trading_DS[purch_doc],Invoice_DP[PO New])

RETURN

IF(Invoice_DP[PO Type] = "PO Trading" && _GR_Trading > 0,_GR_Trading,
IF(_GR > 0,  _GR,
IF(_GR_Project > 0,_GR_Project)))
```

Gambar 3.34 Kode DAX GR Qty

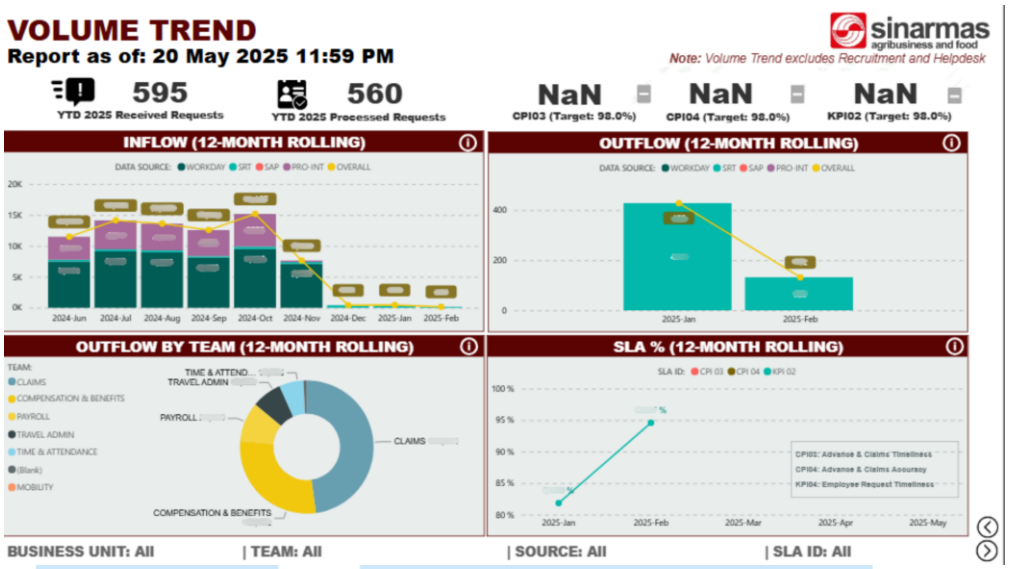
Selain itu, intern juga mengembangkan kode DAX bernama GR Qty, yang memiliki struktur serupa untuk menghitung jumlah barang yang telah diterima. Perhitungan ini mengambil data dari kolom *GR qty* dan *gr_quantity* pada tabel sumber yang relevan, dan mengutamakan hasil *lookup* yang tidak kosong dan memiliki nilai lebih besar dari nol. Logika pengambilan nilai ini disusun secara hierarkis agar akurat dan mencerminkan prioritas sumber data.

```
OS GR QTY = Invoice_DP[PO Qty]-Invoice_DP[GR Qty]
```

Gambar 3.35 Kode DAX OS GR QTY

Terakhir, intern membuat perhitungan OS GR QTY, yaitu selisih antara kuantitas pesanan dan kuantitas barang yang telah diterima, dengan menggunakan rumus Invoice_DP[PO Qty] - Invoice_DP[GR Qty].

3.2.2.8. Update data source dan formula in the visual untuk HR-BP Dashboard

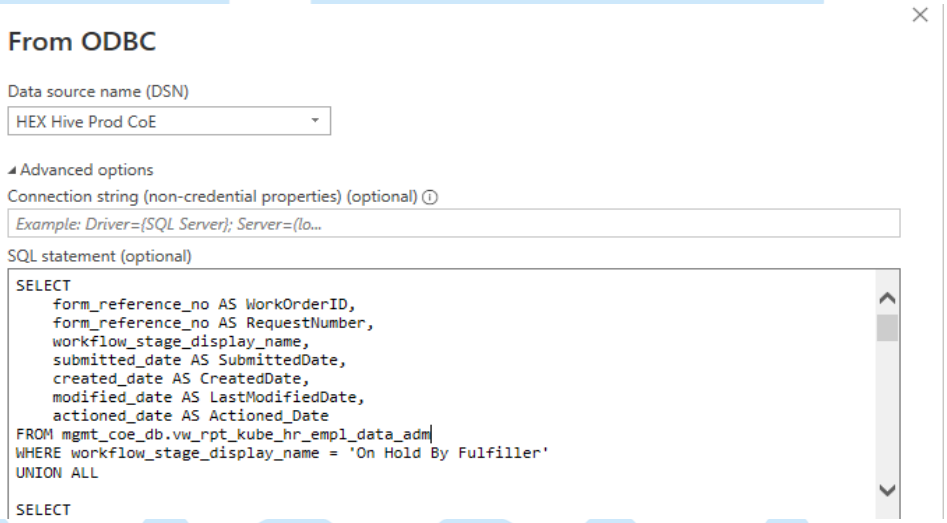


Gambar 3.36 Dashboard HR-BP

Dashboard HR-BP (Human Resources – Business Process) merupakan sebuah alat visualisasi data yang dirancang untuk membantu tim HR Central Business Services (CBS) dalam memantau dan mengevaluasi efektivitas proses layanan karyawan. Tujuan utama dari dashboard ini adalah memberikan gambaran menyeluruh mengenai volume permintaan layanan karyawan dan ketepatan waktu penyelesaiannya berdasarkan standar yang telah ditentukan dalam Service Level Agreement (SLA). Melalui dashboard ini, pengguna dapat melihat sejauh mana permintaan seperti pengajuan data, dokumen, maupun laporan karyawan diselesaikan sesuai dengan waktu yang ditargetkan, serta mengidentifikasi kendala yang menyebabkan keterlambatan. Dashboard ini juga memungkinkan tim HR CBS untuk melakukan analisis berbasis data terhadap penyebab SLA tidak tercapai, baik dari sisi sumber permintaan maupun jenis layanannya.

Permasalahan pertama pada dashboard HR-BP yang dikerjakan pada tanggal 25 Februari sampai 28 Februari, terletak pada tabel *RepSrtHR*, khususnya dalam proses perhitungan Turn Around Time (TAT) untuk kasus dengan status *"On Hold by Fulfiller"*. Dalam logika yang digunakan untuk menentukan kolom *Final Status*, data dengan status ini dikategorikan sebagai *"Open"* karena tidak memiliki

Completion Date. Hal ini menyebabkan sistem tetap menghitung *Age* atau lama waktu proses hingga hari ini, dengan logika yang menghitung seluruh hari kerja (workday) dari tanggal *Submit* hingga tanggal sekarang. Namun, dalam praktiknya, status *"On Hold by Fulfiller"* merepresentasikan kondisi di mana proses sedang ditunda karena menunggu kelengkapan atau tindak lanjut dari pihak lain, sehingga tidak seharusnya dihitung dalam waktu aktif penyelesaian layanan. Akibatnya, *Age* terus bertambah meskipun proses sebenarnya sedang tertunda, dan bahkan menghitung hari kerja secara terus-menerus, termasuk melewati akhir pekan secara tidak tepat. Hal ini menimbulkan distorsi dalam pengukuran SLA dan menurunkan akurasi analisis kinerja layanan. Oleh karena itu, diperlukan penyesuaian pada logika perhitungan agar *Age* untuk status *"On Hold by Fulfiller"* tidak terus bertambah atau mengecualikan waktu saat status tersebut aktif



Gambar 3.37 Logic SQL untuk pengambilan data stage "On Hold By Fulfiller"

Untuk mengatasi permasalahan perhitungan *Turn Around Time (TAT)* pada status *"On Hold By Fulfiller"*, intern melakukan langkah awal dengan membuat *query* baru (Query3) yang mengambil data dari enam tabel sumber terkait proses HR, seperti *empl_data_adm*, *benefits*, dan *personal_claims*. Query ini secara spesifik hanya mengambil data dengan status *"On Hold By Fulfiller"* untuk merekam periode waktu ketika suatu permintaan dalam kondisi tertahan.

```
DifferenceDays =
VAR StartDate = DATEVALUE(Query3[CreatedDate])
VAR EndDate = DATEVALUE(Query3[Actioned_Date])
VAR DateList =
    FILTER(
        CompletionCalendar,
        CompletionCalendar[Date] >= StartDate &&
        CompletionCalendar[Date] <= EndDate &&
        CompletionCalendar[IsWorkDay] = 1
    )

VAR WorkdayCount = COUNTROWS(DateList)

RETURN
    WorkdayCount
```

Gambar 3.38 Kode DAX DifferenceDays

Setelah data berhasil dikumpulkan, intern membuat logika tambahan dalam bentuk *DAX* untuk menghitung selisih hari kerja antara *CreatedDate* dan *Actioned_Date*. Kalkulasi dilakukan dengan memanfaatkan tabel referensi *CompletionCalendar* yang sudah menyaring hanya hari kerja (weekday) dengan nilai IsWorkDay = 1. Hasil perhitungan ini disimpan dalam kolom DifferenceDays.

```
TotalWorkdays =
VAR CurrentWorkOrderID = Query3[WorkOrderID]
VAR TotalWorkdays =
    CALCULATE(
        SUM(Query3[DifferenceDays]),
        FILTER(Query3, Query3[WorkOrderID] = CurrentWorkOrderID)
    )
RETURN
    TotalWorkdays
```

Gambar 3.39 Kode DAX TotalWorkdays

Langkah berikutnya adalah menghitung total hari kerja untuk setiap permintaan dengan menjumlahkan DifferenceDays berdasarkan *WorkOrderID*. Hasil akhir disimpan dalam kolom TotalWorkdays, yang dapat digunakan untuk mengecualikan durasi *on-hold* dari total perhitungan TAT.

```
TAT_no_On_Hold =
VAR CurrentWorkOrderID = RepSrtHR[WorkOrderID]
VAR MaxTotalWorkdays =
    CALCULATE(
        MAX(Query3[TotalWorkdays]),
        FILTER(Query3, Query3[WorkOrderID] = CurrentWorkOrderID)
    )
RETURN
    IF(
        ISBLANK(MaxTotalWorkdays),
        RepSrtHR[TAT],
        RepSrtHR[TAT] - MaxTotalWorkdays
    )
```

Gambar 3.40 Kode DAX TAT_no_On_Hold

Setelah berhasil menghitung total hari kerja status "On Hold By Fulfiller", intern melanjutkan dengan mengembangkan metrik baru bernama TAT_no_On_Hold. Kolom ini menghitung selisih Turn Around Time (TAT) setelah mengecualikan durasi hari kerja saat permintaan berada dalam status on hold. Jika data WorkOrderID tidak ditemukan dalam referensi, maka nilai TAT tidak dikurangi.

```
SLM_STatus2 =
IF (
    RepSrtHR[Source] = "SRT", RepSrtHR[SLMStatus],
    IF(
        RepSrtHR[TAT_no_On_Hold]>5 , "Service Target Breached","Within the Service Target"))
```

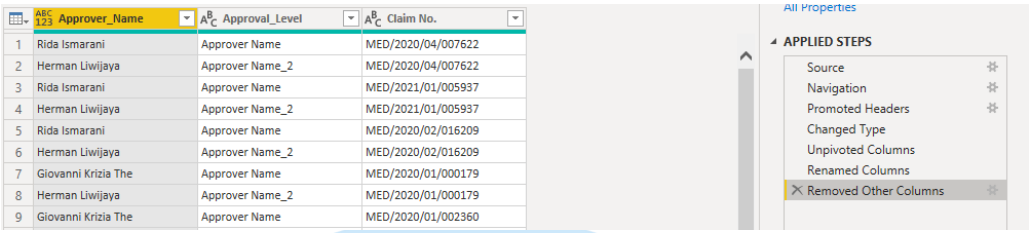
Gambar 3.41 Kode DAX SLM_Status2

Selanjutnya, kolom SLM_Status2 dibuat untuk menetapkan status SLA berdasarkan nilai TAT_no_On_Hold, dengan ketentuan jika melebihi 5 hari kerja maka dikategorikan sebagai Service Target Breached, dan jika tidak, dianggap Within the Service Target.

```
SLA =
IF(
    (RepSrtHR[Final Status]="Open" || RepSrtHR[SLA ID]=BLANK() || RepSrtHR[Final Status]="Cancelled" || RepSrtHR[Final Status]="Rejected" ),
    BLANK(),
        IF(
            RepSrtHR[SLA ID]="CPI 04", "Miss",
                IF(RepSrtHR[SLM_STatus2]="Within the Service Target",
                    "Pass","Miss")))
```

Gambar 3.42 Kode DAX SLA

Terakhir, metrik SLA digunakan sebagai indikator utama keberhasilan SLA, dengan mempertimbangkan kondisi status akhir, kejelasan SLA ID, serta hasil evaluasi SLM_Status2.

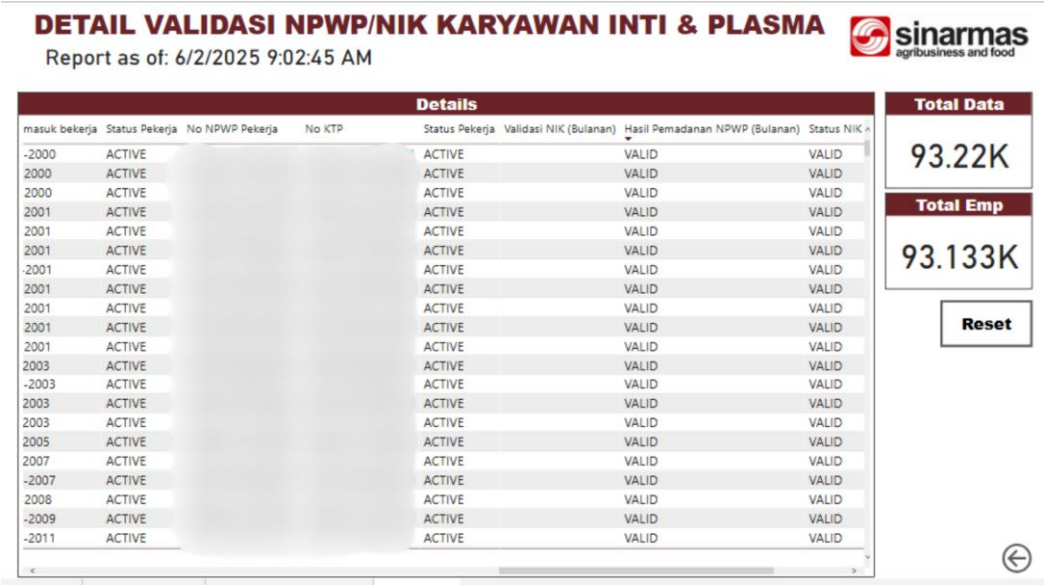


	Approver_Name	Approval_Level	Claim No.
1	Rida Ismarani	Approver Name	MED/2020/04/007622
2	Herman Liwijaya	Approver Name_2	MED/2020/04/007622
3	Rida Ismarani	Approver Name	MED/2021/01/005937
4	Herman Liwijaya	Approver Name_2	MED/2021/01/005937
5	Rida Ismarani	Approver Name	MED/2020/02/016209
6	Herman Liwijaya	Approver Name_2	MED/2020/02/016209
7	Giovanni Krizia The	Approver Name	MED/2020/01/000179
8	Herman Liwijaya	Approver Name_2	MED/2020/01/000179
9	Giovanni Krizia The	Approver Name	MED/2020/01/002360

Gambar 3.43 Tampilan Steps Unpivoted Column Approver

Permasalahan kedua yang ditemukan pada pengolahan data dashboard HR-BP berkaitan dengan struktur data pada kolom *approver*, di mana satu baris data memuat beberapa nama *approver* sekaligus. Struktur ini menyulitkan proses analisis, khususnya ketika ingin melakukan penyaringan atau perhitungan jumlah dokumen yang telah disetujui oleh masing-masing *approver*. Hal ini disebabkan karena sistem tidak dapat memfilter setiap nama *approver* secara individual apabila digabung dalam satu kolom. Untuk mengatasi permasalahan tersebut, dilakukan transformasi data menggunakan teknik *unpivot column*. Dengan teknik ini, setiap nama *approver* yang sebelumnya berada dalam satu baris dijadikan baris terpisah berdasarkan dokumen yang sama. Hasil dari transformasi ini memungkinkan setiap *approver* dapat dianalisis secara individual, sehingga proses penyaringan dan penghitungan jumlah dokumen yang telah disetujui oleh masing-masing pihak menjadi lebih akurat dan terstruktur.

3.2.2.9. Update data dan logic DAX dashboard "Master Data Karyawan”



Gambar 3.44 Tampilan dashboard Master Data Karyawan

Dashboard FA – Master Data Karyawan dirancang untuk memberikan visibilitas terhadap status kelengkapan dan validitas data Nomor Pokok Wajib Pajak (NPWP) dan Nomor Induk Kependudukan (NIK) bagi karyawan Non-Staff berdasarkan sumber data Checkroll dan NCIP. Tujuan utama dari dashboard ini adalah untuk membantu tim administrasi dalam memantau dan memperbaiki kelengkapan data identitas karyawan, yang dikelompokkan berdasarkan status validitas serta wilayah kerja, seperti kelompok inti dan plasma. Status validasi NIK dan NPWP diperoleh melalui mekanisme *lookup* terhadap hasil pemadanan data yang tersedia.

Logika klasifikasi data dalam dashboard ini terbagi ke dalam empat kategori. Pertama, apabila Status NIK bernilai *Valid* atau Status NPWP menunjukkan hasil *Padan*, maka karyawan dikategorikan sebagai “Karyawan dengan NIK Valid”. Kedua, apabila Status NIK bernilai *Invalid*, maka dikategorikan sebagai “Karyawan dengan NIK Tidak Valid”. Ketiga, jika Status NIK kosong dan Status NPWP bernilai *Tidak Padan*, maka juga dikategorikan sebagai “Karyawan dengan NIK Tidak Valid”. Keempat, apabila tidak memenuhi ketiga kondisi tersebut, maka dikategorikan sebagai “Karyawan Tanpa Data NIK”.

Gambar 3.45 Tabel Validasi NIK dan NPWP baru

Selama pelaksanaan tugas pada dashboard FA – Master Data Karyawan, intern melakukan pembaruan data dengan menambahkan dua sumber validasi baru, yaitu tabel Validasi NPWP – DJP dan Validasi NIK – DJP. Penambahan sumber ini bertujuan untuk meningkatkan akurasi pemadanan data identitas karyawan dengan data resmi dari Direktorat Jenderal Pajak (DJP), sehingga proses verifikasi NIK dan NPWP menjadi lebih andal dan terstandarisasi.

```
NPWP Pekerja Clean =
VAR Original = 'Validasi NPWP - DJP'[No NPWP Pekerja]
VAR RemovedDots = SUBSTITUTE(Original, ".", "")
VAR RemovedDashes = SUBSTITUTE(RemovedDots, "-", "")
RETURN
    RemovedDashes
```

Gambar 3.46 Kode DAX NPWP Pekerja Clean

```
Status Valid 2 =
IF ( 'Validasi NPWP - DJP'[STATUS_VALIDASI] = "Error" ,"INVALID" , "VALID")
```

Gambar 3.10 Informasi permasalahan dashboard dari team procurement

```
Nama_NPWP =
'Validasi NPWP - DJP'[Nama Pekerja] & " " &
'Validasi NPWP - DJP'[NPWP Pekerja Clean]
```

Gambar 3.47 Kode DAX Nama_NPWP

Pada tabel Validasi NPWP – DJP, dilakukan pembersihan format data NPWP melalui kalkulasi NPWP Pekerja Clean, dengan cara menghilangkan karakter titik (.) dan tanda hubung (-) dari kolom *No NPWP Pekerja*. Setelah pembersihan, status validasi ditentukan melalui logika Status Valid 2, di mana jika kolom *STATUS_VALIDASI* bernilai *Error*, maka status dianggap *INVALID*, selain itu dianggap *VALID*. Kemudian, dibentuk kolom Nama_NPWP sebagai gabungan antara nama pekerja dan NPWP yang telah dibersihkan untuk mempermudah proses pencocokan antar tabel.

```
NIK Pekerja Clean =
VAR Original = 'Validasi NIK - DJP'[No KTP]
VAR RemovedDots = SUBSTITUTE(Original, ".", "")
VAR RemovedDashes = SUBSTITUTE(RemovedDots, "-", "")
VAR RemovedBackslash = SUBSTITUTE(RemovedDashes, "\", "")
VAR RemovedSingleQuote = SUBSTITUTE(RemovedBackslash, "'", "")
RETURN
    RemovedSingleQuote
```

Gambar 3.48 Kode DAX NIK Pekerja Clean

```
Nama_NIK =
'Validasi NIK - DJP'[Nama Pekerja] & " " &
'Validasi NIK - DJP'[NIK Pekerja Clean]
```

Gambar 3.49 Kode DAX Nama_NIK

Sementara itu, dalam tabel Validasi NIK – DJP, proses validasi dilakukan dengan logika serupa. Pembersihan NIK dilakukan pada kolom NIK Pekerja Clean, yang menghilangkan karakter tidak relevan seperti titik (.), tanda hubung (-), tanda backslash (\), dan tanda kutip satu ('). Selanjutnya, dibentuk kolom Nama_NIK sebagai kombinasi antara nama pekerja dan NIK yang telah dibersihkan. Setelah proses pembersihan data pada kedua tabel validasi, yaitu Validasi NPWP – DJP dan Validasi NIK – DJP, langkah selanjutnya adalah mengintegrasikan hasil pembersihan dan validasi tersebut ke dalam tabel utama yang dinamakan Master Data. Integrasi ini bertujuan untuk memperbarui status validasi pada setiap entri karyawan berdasarkan sumber validasi terbaru.

```
Status NIK =
VAR Validasi =  LOOKUPVALUE('Validasi NIK - DJP'[STATUS_VALIDASI],'Validasi NIK - DJP'[Nama_NIK], 'Master Data'[Nama_NIK])
VAR Length = LEN('Master Data'[NIK Pekerja Clean])
RETURN
IF(Validasi = BLANK(), BLANK(), Validasi)
```

Gambar 3.50 Kode DAX Status NIK

Kolom Status NIK diperbarui dengan menggunakan fungsi LOOKUPVALUE, yang mencocokkan nilai *Nama_NIK* pada Master Data dengan kolom yang sama pada tabel Validasi NIK – DJP, dan mengambil nilai dari kolom *STATUS_VALIDASI*. Jika tidak ditemukan kecocokan, maka hasilnya kosong (*blank*); jika ditemukan, akan ditampilkan status *VALID* atau *INVALID* sesuai hasil validasi dari DJP.

```
Status NPWP =
VAR Validasi = LOOKUPVALUE('Validasi NPWP - DJP'[STATUS_VALIDASI],'Validasi NPWP - DJP'[Nama_NPWP], 'Master Data'[Nama_NPWP])
VAR Length = LEN('Master Data'[NPWP Pekerja Clean])

RETURN
IF(Validasi = BLANK(), BLANK(), Validasi)
```

Gambar 3.51 Kode DAX Status NPWP

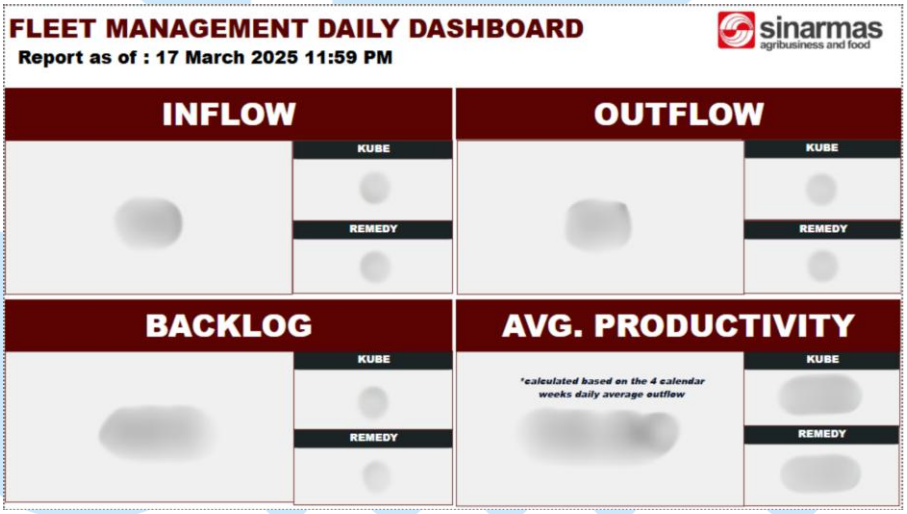
Hal serupa dilakukan pada kolom Status NPWP, yang mengambil hasil validasi dari tabel Validasi NPWP – DJP berdasarkan kecocokan nilai *Nama_NPWP*. Fungsi LOOKUPVALUE digunakan untuk mencocokkan data dan mengambil status validasinya.

```
Status Akhir =
IF
('Master Data'[Status NIK] = "VALID" ||
'Master Data'[Status NPWP] = "VALID" , "VALID" , "INVALID")
```

Gambar 3.52 Kode DAX Status Akhir

Selanjutnya, dibuat kolom Status Akhir untuk menyimpulkan kondisi data identitas. Jika *Status NIK* atau *Status NPWP* bernilai *VALID*, maka *Status Akhir* akan bernilai *VALID*. Jika keduanya tidak valid atau tidak tersedia, maka akan diberi nilai *INVALID*.

3.2.2.10. Membuat formula filter date DAX GA Daily Dashboard



Gambar 3.53 Tampilan dashboard DAX GA Daily

Perbaikan yang dilakukan pada GA Daily Dashboard berfokus pada penyederhanaan proses filterisasi data, khususnya untuk elemen-elemen visual seperti *Inflow*, *Outflow*, *Backlog*, dan *Average Product*. Sebelumnya, setiap

visualisasi memiliki filter tanggal tersendiri sehingga pengguna harus melakukan penyaringan data secara manual satu per satu. Pembaruan ini mengintegrasikan seluruh filter tanggal ke dalam satu kontrol pusat, sehingga seluruh visualisasi dapat menyesuaikan secara otomatis berdasarkan satu pilihan tanggal saja. Implementasi pembaruan ini diterapkan ke dalam sepuluh dashboard GA Daily, yang meliputi: GA Admin, Asset Management, Guest Housing, IWS, Property Management, Fleet Management, Helpdesk, Office and Facility, serta Document Tracking.

```
Filter Calendar =
ADDCOLUMNS (
CALENDAR (DATE(2018 , 1 , 1 ), date(2030,1,1)),
"DateAsInteger", FORMAT ( [Date], "YYYYMMDD" ),
"Year", YEAR ( [Date] ),
"Monthnumber", FORMAT ( [Date], "MM" ),
"YearMonthnumber", FORMAT ( [Date], "YYYY/MM" ),
"YearMonthShort", FORMAT ( [Date], "YYYY-mmm" ),
"MonthNameShort", FORMAT ( [Date], "mmm" ),
"MonthNameLong", FORMAT ( [Date], "mmmm" ),
"DayOfWeekNumber", WEEKDAY ( [Date] ),
"DayOfWeek", FORMAT ( [Date], "dddd" ),
"DayOfWeekShort", FORMAT ( [Date], "ddd" ),
"Quarter", "Q" & FORMAT ( [Date], "Q" ),
"YearQuarter", FORMAT ( [Date], "YYYY" ) & "/Q" & FORMAT ( [Date], "Q" )
)
```

Gambar 3.54 Kode DAX tabel Filter Calendar

Untuk melakukan update pada GA Daily Dashboard, langkah pertama yang dilakukan oleh intern adalah membuat tabel baru bernama Filter Calendar. Tabel ini berfungsi sebagai acuan filter tanggal terpadu yang akan digunakan untuk semua visualisasi dalam dashboard. Tabel ini dibuat dengan menggunakan fungsi CALENDAR, mencakup rentang tanggal dari 1 Januari 2018 hingga 1 Januari 2030, dan dilengkapi dengan berbagai kolom tambahan seperti *Year*, *Month*, *Quarter*, hingga *Day of Week* untuk mendukung kebutuhan filterisasi lanjutan.

```
#Inflow_helpdesk =
VAR SelectedDate= [Filter Date]

VAR OriginalDate =
CALCULATE(DISTINCTCOUNT('GF, BC, AT&M - KUBE'[form_reference_no]),KEEPFILTERS('GF, BC, AT&M - KUBE'[workflow_stage_display_name]="In Progress by Fulfiller")) + CALCULATE(DISTINCTCOUNT('GF, BC, AT&M - RepSrtGA'[WorkOrderID]))+0

RETURN
IF (SelectedDate = BLANK(), OriginalDate,
CALCULATE(DISTINCTCOUNT('GF, BC, AT&M - KUBE'[form_reference_no]),KEEPFILTERS('GF, BC, AT&M - KUBE'[workflow_stage_display_name]="In Progress by Fulfiller"), FILTER(CalendarInflow, CalendarInflow[Date]= SelectedDate)) + CALCULATE(DISTINCTCOUNT('GF, BC, AT&M - RepSrtGA'[WorkOrderID]),FILTER(CalendarInflow, CalendarInflow[Date]= SelectedDate))+0)
```

Gambar 3.55 Kode DAX #Inflow_helpdesk

```
#Outflow_helpdesk =
VAR SelectedDate= [Filter Date]

VAR OriginalDate =
CALCULATE(DISTINCTCOUNT('GF, BC, AT&M - KUBE'[form_reference_no]), KEEPFILTERS('GF, BC, AT&M - KUBE'[form_status] IN {"Completed","Fulfiller Rejected"})) +
CALCULATE(DISTINCTCOUNT('GF, BC, AT&M - RepSrtGA'[WorkOrderID]), KEEPFILTERS('GF, BC, AT&M - RepSrtGA'[Final Status]<>"Open")) + 0

RETURN
IF (SelectedDate = BLANK(), OriginalDate,
CALCULATE(DISTINCTCOUNT('GF, BC, AT&M - KUBE'[form_reference_no]), KEEPFILTERS('GF, BC, AT&M - KUBE'[form_status] IN {"Completed","Fulfiller Rejected"}),FILTER(CalendarOutflow, CalendarOutflow[Date]= SelectedDate)) +
CALCULATE(DISTINCTCOUNT('GF, BC, AT&M - RepSrtGA'[WorkOrderID]), KEEPFILTERS('GF, BC, AT&M - RepSrtGA'[Final Status]<>"Open"),FILTER(CalendarOutflow, CalendarOutflow[Date]= SelectedDate)) + 0)
```

Gambar 3.56 Kode DAX #Outflow_helpdesk

```
#Backlog - KPI 06 Helpdesk =
VAR SelectedDate = [Filter Date]
RETURN
CALCULATE(
    DISTINCTCOUNT('GF, BC, AT&M - KUBE'[form_reference_no]),
    FILTER(
        'GF, BC, AT&M - KUBE',
        'GF, BC, AT&M - KUBE'[in-date date] <= SelectedDate &&
        (ISBLANK('GF, BC, AT&M - KUBE'[out-date]) || 'GF, BC, AT&M - KUBE'[out-date] > SelectedDate) &&
        'GF, BC, AT&M - KUBE'[form_status] <> "Cancelled" && 'GF, BC, AT&M - KUBE'[form_status] <> "Pending Requestor"
    )) +
CALCULATE(
    DISTINCTCOUNT('GF, BC, AT&M - RepSrtGA'[WorkOrderID]),
    FILTER(
        'GF, BC, AT&M - RepSrtGA',
        'GF, BC, AT&M - RepSrtGA'[Submit Date] <= SelectedDate &&
        (ISBLANK('GF, BC, AT&M - RepSrtGA'[Completion Date]) || 'GF, BC, AT&M - RepSrtGA'[Completion Date] > SelectedDate) &&
        'GF, BC, AT&M - RepSrtGA'[Status] <> "Cancelled" && 'GF, BC, AT&M - RepSrtGA'[Status] <> "Pending Requestor"
    )) + 0
```

Gambar 3.57 Kode DAX #Backlog - KPI 06 Helpdesk

```
#AvgProd_Helpdesk =
VAR SelectedDate = SELECTEDVALUE('Filter Calendar'[Date], MAX(CalendarOutflow[Date]))
VAR Last4Weeks =
    DATESINPERIOD(CalendarOutflow[Date], SelectedDate, -28, DAY)
VAR TotalOutflowLast4Weeks =
    CALCULATE(
        SUMX(
            VALUES(CalendarOutflow[Date]),
            [#Outflow_helpdesk2]
        ), Last4Weeks)
VAR TotalWorkingDays =
    CALCULATE(
        SUM(CalendarOutflow[IsWorkingDay]),
        Last4Weeks)
RETURN
    IF(TotalWorkingDays = 0, 0, TotalOutflowLast4Weeks / TotalWorkingDays)
```

Gambar 3.58 Kode DAX #AvgProd_Helpdesk

Setelah tabel Filter Calendar dibuat, langkah berikutnya adalah menerapkan logic baru pada setiap visualisasi utama dalam dashboard, yaitu *Inflow*, *Outflow*, *Backlog*, dan *Average Product*, agar semuanya merujuk pada satu sumber tanggal dari Filter Calendar. Sebagai contoh, dalam dashboard Office & Facility, visualisasi *Inflow* dihitung berdasarkan status proses dari dua sumber data utama dan akan menyesuaikan hasilnya jika pengguna memilih tanggal tertentu melalui filter. Logic serupa juga diterapkan pada *Outflow* dengan status penyelesaian data, *Backlog* dengan mempertimbangkan entri yang masih dalam proses berdasarkan tanggal masuk dan keluar, serta *Average Product* yang dihitung berdasarkan rata-rata output selama 28 hari kerja terakhir.

Seluruh formula tersebut memanfaatkan variabel SelectedDate yang mengambil input dari tabel filter tanggal, serta memperhitungkan kondisi apabila tidak ada tanggal yang dipilih. Dengan pendekatan ini, proses analisis menjadi lebih konsisten dan efisien karena pengguna cukup melakukan satu kali filter untuk melihat seluruh data dari berbagai visualisasi secara terpadu. Update ini diaplikasikan ke sepuluh dashboard GA Daily, termasuk GA Admin, Asset Management, Guest Housing, IWS, Property Management, Fleet Management, Helpdesk, Office and Facility, dan Document Tracking.

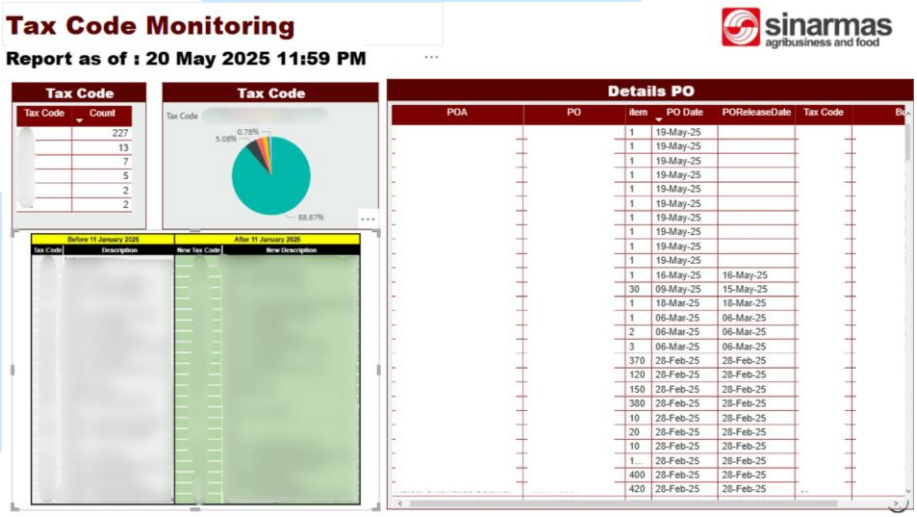
Summary Total & Average Dashboard 01-28 Februari 2025

Date	GA Admin				Asset Management				Guest Housing				IWS				Property Management				
	Inflow	Outflow	Backlog	Avg. Product	Inflow	Outflow	Backlog	Avg. Product	Inflow	Outflow	Backlog	Avg. Product	Inflow	Outflow	Backlog	Avg. Product	Inflow	Outflow	Backlog	Avg. Product	Inflow
01/02/2025	0	0	9	5,06	0	0	0	1,83	1	0	1	3	0	0	2	1	0	0	0	27,94	0
02/02/2025	0	0	9	5,11	0	0	0	1,83	2	1	1	3	0	0	2	1	0	0	0	27,94	0
03/02/2025	5	0	14	5,11	0	0	0	1,83	20	10	11	3	4	6	4	1	11	11	0	27,94	23
04/02/2025	3	8	9	5,11	0	0	0	1,83	1	6	0	3	1	1	5	1	39	38	0	27,94	18
05/02/2025	4	26	12	5,11	0	0	0	1,83	6	17	0	3	1	6	1	1	32	32	0	27,94	22
06/02/2025	11	5	15	5,06	0	0	0	1,83	1	1	1	3	1	1	1	1	31	31	0	27,94	14
07/02/2025	2	2	12	5,06	2	2	0	1,83	6	1	0	3	1	2	0	1	36	35	0	28	19
08/02/2025	0	0	12	5,78	0	0	0	1,94	1	1	0	4	0	0	0	1	0	0	0	28,28	4
09/02/2025	0	0	12	5,94	0	0	0	1,94	0	1	0	4	0	0	0	1	0	0	0	28,28	0
10/02/2025	2	1	15	5,94	0	0	0	1,94	3	1	1	4	2	2	2	1	25	25	0	28,28	13
11/02/2025	5	6	12	5,94	0	0	0	1,94	4	3	0	4	1	3	0	1	15	13	0	28,28	26
12/02/2025	6	5	12	5,94	0	0	0	1,94	5	1	0	4	0	0	0	1	27	25	1	28,28	27
13/02/2025	5	0	17	5,94	0	0	0	1,94	2	5	0	4	0	0	0	1	28	28	1	28,28	18
14/02/2025	7	4	18	5,78	0	0	0	1,94	6	8	0	4	2	2	2	1	47	47	1	28,28	16
15/02/2025																					
16/02/2025	0	0	18	5,56	0	0	0	1,94	0	1	0	4	0	0	2	1	0	0	1	28,83	1
17/02/2025	5	6	17	5,56	1	0	0	1,94	7	12	1	4	2	2	4	1	34	34	1	28,83	16
18/02/2025	6	4	19	5,56	2	0	0	1,94	6	40	0	4	1	2	4	1	33	33	0	28,83	15

Gambar 3.59 Tabel validasi excel GA Daily

Selama proses pembaruan logika pada masing-masing visualisasi dalam dashboard GA Daily, intern juga melakukan tahapan validasi hasil. Validasi ini dilaksanakan dengan membandingkan keluaran dari dashboard yang telah diperbarui dengan file Excel referensi yang disediakan oleh pengguna dari tim GA. File Excel tersebut memuat data historis atau rekapitulasi resmi yang digunakan sebagai acuan. Melalui proses ini, intern memastikan bahwa jumlah *Inflow*, *Outflow*, *Backlog*, hingga *Average Product* yang ditampilkan pada dashboard telah sesuai dan akurat, serta mencerminkan kondisi operasional yang sebenarnya.

3.2.2.11. Meningkatkan dan memperbaiki Procurement Purchase Order Dashboard

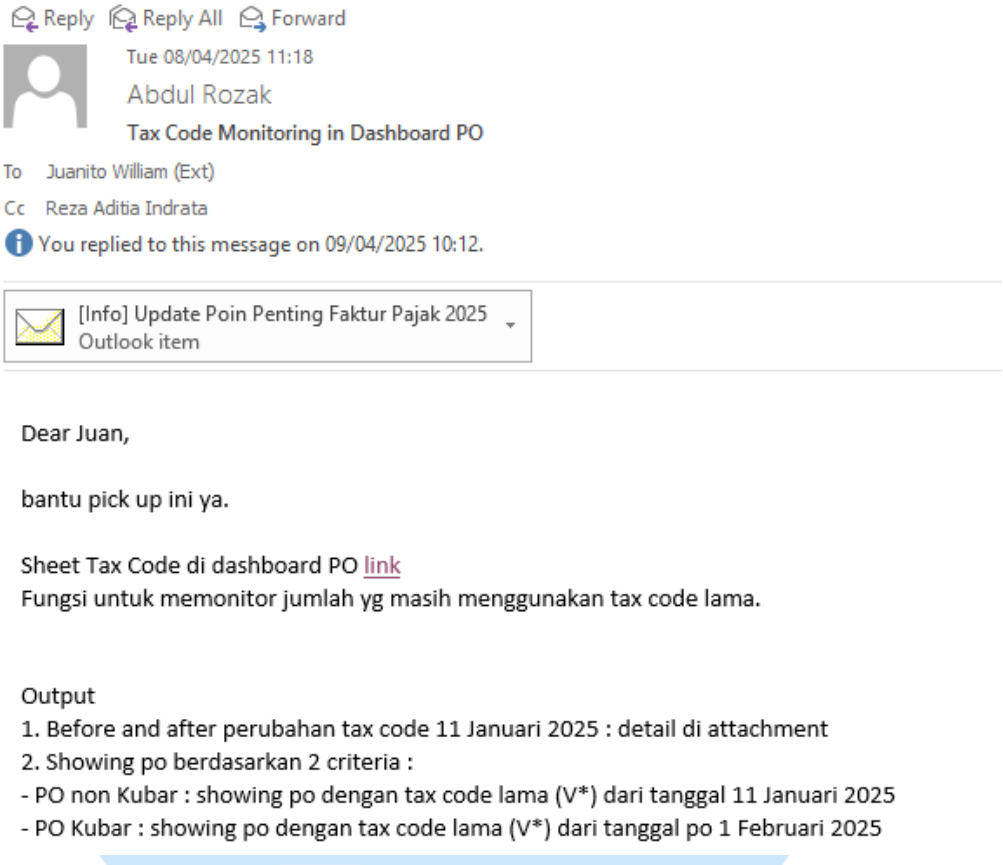


Gambar 3.60 Tampilan dashboard Procurement Purchase Order

Dashboard selanjutnya yang saya tingkatkan adalah dashboard Procurement Purchase Order, dengan fokus utama pada pembaruan logika dan penyempurnaan

filter agar selaras dengan kebijakan perubahan *tax code* yang berlaku di tahun 2025. Pembaruan Dashboard Procurement Purchase Order Berdasarkan Kebijakan PPN 2025 Dashboard *Procurement Purchase Order* diperbarui sebagai bentuk respons terhadap perubahan regulasi perpajakan yang mulai berlaku pada 1 Januari 2025. Perubahan ini merujuk pada Peraturan Menteri Keuangan Nomor 131 Tahun 2024 serta amanat dari Undang-Undang Nomor 7 Tahun 2021 tentang Harmonisasi Peraturan Perpajakan (UU HPP), yang menetapkan kenaikan tarif Pajak Pertambahan Nilai (PPN) menjadi 12%. Meski secara nominal meningkat, kebijakan ini tetap mempertahankan beban pajak yang ringan bagi masyarakat karena dasar pengenaan pajaknya disesuaikan menjadi 11/12 dari harga jual. Dalam praktiknya, perubahan tarif dan dasar pengenaan pajak ini juga berdampak pada penggunaan dan pengkodean tax code (kode pajak) dalam sistem pembelian perusahaan, khususnya dalam proses pembuatan Purchase Order (PO).

Untuk menyesuaikan dengan kebijakan tersebut dan memudahkan tim Procurement dalam melakukan monitoring, dashboard diperbarui dengan dua fitur utama: (1) penambahan tabel baru yang memperlihatkan kondisi *before* dan *after* dari penggunaan tax code, dan (2) penerapan filter dinamis berdasarkan logika waktu dan wilayah kerja (Kubar vs. non-Kubar). Pembaruan ini bertujuan untuk memastikan bahwa seluruh proses PO yang dilakukan setelah tanggal 11 Januari 2025 (atau 1 Februari 2025 khusus untuk unit Kubar) telah menggunakan tax code yang sesuai dengan kebijakan pemerintah. Dengan pembaruan ini, dashboard tidak hanya menjadi alat pelaporan, tetapi juga alat pengawasan terhadap kepatuhan proses bisnis terhadap regulasi perpajakan terbaru. Dalam proses pembaruan ini, saya melakukan dua tugas utama sesuai dengan request user yang dapat dilihat pada gambar 3.66.


Reply Reply All Forward

Tue 08/04/2025 11:18

Abdul Rozak

Tax Code Monitoring in Dashboard PO

To Juanito William (Ext)

Cc Reza Aditia Indrata

You replied to this message on 09/04/2025 10:12.

[Info] Update Poin Penting Faktur Pajak 2025
Outlook item

Dear Juan,

bantu pick up ini ya.

Sheet Tax Code di dashboard PO link
Fungsi untuk memonitor jumlah yg masih menggunakan tax code lama.

Output
1. Before and after perubahan tax code 11 Januari 2025 : detail di attachment
2. Showing po berdasarkan 2 criteria :
- PO non Kubar : showing po dengan tax code lama (V*) dari tanggal 11 Januari 2025
- PO Kubar : showing po dengan tax code lama (V*) dari tanggal po 1 Februari 2025

Gambar 3.61 Request perbaikan dashboard oleh user

Pertama, membuat tabel baru yang memuat data *before* dan *after* perubahan *tax code* per tanggal 11 Januari 2025 dilakukan sebagai bentuk penyesuaian terhadap peraturan pemerintah terkait perubahan tarif dan pengkodean *Value Added Tax* (VAT). Kebijakan ini berdampak langsung pada proses transaksi pembelian di perusahaan, khususnya dalam penggunaan *tax code* pada setiap *purchase order* (PO). Oleh karena itu, tugas ini bertujuan untuk membantu tim Procurement dalam memantau dan membandingkan kondisi PO sebelum dan sesudah kebijakan diberlakukan.

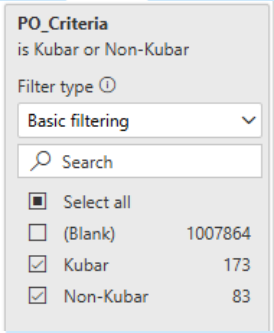
Gambar 3.62 Table validasi baru kode Value Added Tax (VAT)

Sebagai langkah lanjutan, intern menambahkan filter dinamis untuk menampilkan data *purchase order* (PO) berdasarkan dua kriteria utama yang disesuaikan dengan kebijakan perubahan *tax code*. Kriteria pertama ditujukan untuk PO non-Kubar, yaitu hanya menampilkan PO dengan *tax code* lama (kode yang diawali huruf "V") untuk transaksi yang dibuat hingga tanggal 11 Januari 2025. Kriteria kedua ditujukan untuk PO yang berasal dari unit Kubar, di mana *tax code* lama tetap ditampilkan hingga tanggal 1 Februari 2025. Logika ini diterapkan menggunakan ekspresi berikut:

```
PO_Criteria =
IF(
    PO_Details[plant_unit] = "Kubar" &&
    PO_Details[po_creation_date] >= DATE(2025, 2, 1) &&
    LEFT(PO_Details[tax_code], 1) = "V",
    "Kubar",
    IF(
        PO_Details[plant_unit] <> "Kubar" &&
        PO_Details[po_creation_date] >= DATE(2025, 1, 11) &&
        LEFT(PO_Details[tax_code], 1) = "V",
        "Non-Kubar"
    )
)
```

Gambar 3.63 Kode DAX PO_Criteria

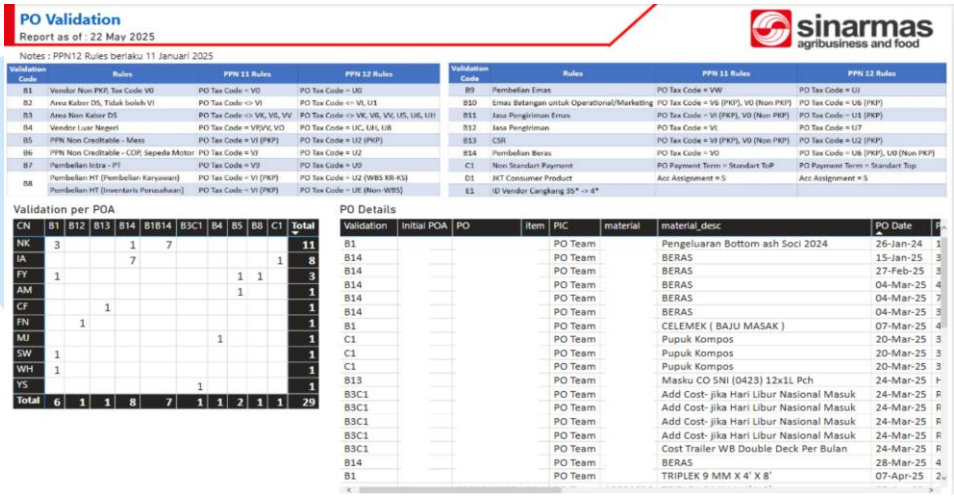
Setelah ekspresi DAX PO_Criteria dibuat, nilai dari kolom ini kemudian dimasukkan ke dalam tab filter pada dashboard. Tujuannya adalah untuk memungkinkan pengguna melakukan penyaringan data PO berdasarkan klasifikasi tersebut, baik untuk PO Kubar maupun non-Kubar. Dengan pengelompokan ini, pengguna dapat dengan mudah membedakan transaksi mana yang masih menggunakan *tax code* lama setelah tanggal yang telah ditentukan.



Gambar 3.64 Filter filter hasil dari kode DAX PO_Criteria

Logika ini dibuat untuk memisahkan dan mengidentifikasi PO yang masih menggunakan *tax code* lama setelah tenggat waktu yang ditentukan oleh pemerintah. Dengan filter ini, pengguna dashboard dapat memantau apakah masih terdapat PO yang belum mengikuti regulasi terbaru sesuai wilayah operasional masing-masing.

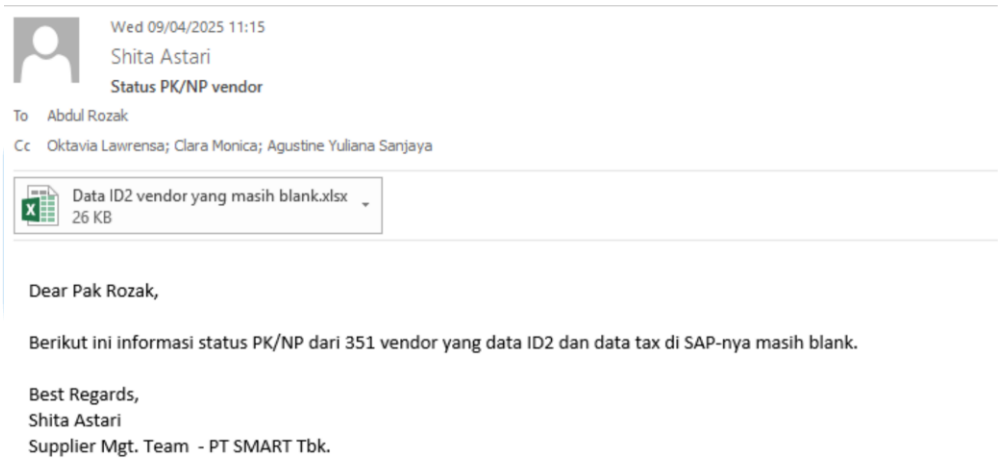
3.2.2.12. Meningkatkan dan memperbaiki Purchase Order Validation Monitoring Dashboard



Gambar 3.65 Tampilan dashboard Purchase Order Validation Monitoring

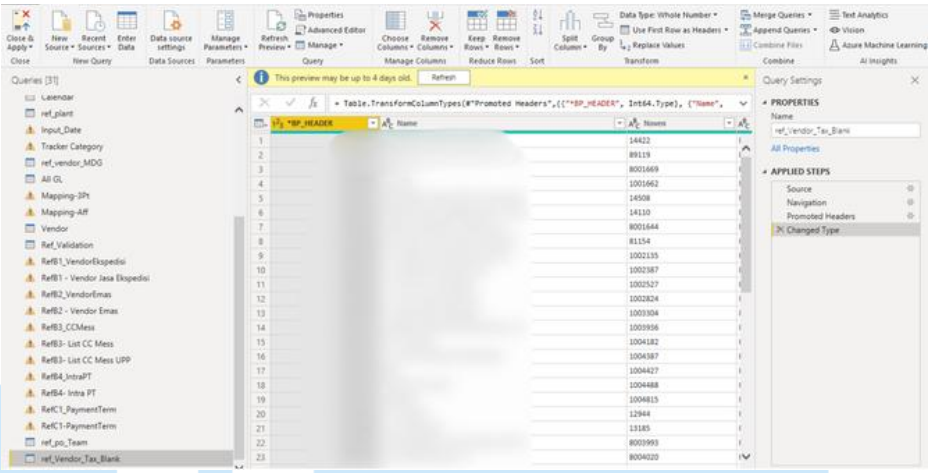
Dashboard Proc-PO-Validation-Monitoring bertujuan untuk memantau dan mengidentifikasi kesalahan dalam penggunaan *Account Assignment* atau kode GL (General Ledger) pada proses pembuatan Purchase Order (PO). Dashboard ini dirancang untuk memastikan bahwa setiap transaksi PO—baik yang melibatkan vendor afiliasi maupun vendor pihak ketiga—menggunakan GL Account yang sesuai dengan jenis transaksinya. Adapun kriteria yang digunakan meliputi seluruh transaksi PO, termasuk identifikasi kesalahan saat transaksi pihak ketiga menggunakan kode vendor afiliasi, serta kesalahan saat transaksi afiliasi menggunakan kode vendor pihak ketiga. Dengan dashboard ini, tim terkait dapat secara cepat mendeteksi dan melakukan koreksi terhadap kesalahan pencatatan yang dapat berdampak pada akurasi laporan keuangan perusahaan.

Permasalahan yang dihadapi dalam proses validasi data vendor adalah adanya kekosongan informasi terkait status pajak pembelian (PK) dan non-pajak (NP) pada data vendor yang digunakan dalam dashboard. Tercatat sebanyak 351 vendor memiliki data yang belum lengkap, terutama pada kolom ID2 dan status tax yang masih kosong pada sistem SAP. Ketidaklengkapan ini menyebabkan dashboard tidak dapat mengidentifikasi secara akurat apakah suatu vendor dikenakan pajak atau tidak, sehingga berpotensi menimbulkan kesalahan dalam pelaporan dan pengambilan keputusan terkait transaksi pembelian.


Wed 09/04/2025 11:15
Shita Astari
Status PK/NP vendor
To Abdul Rozak
Cc Oktavia Lawrensa; Clara Monica; Agustine Yuliana Sanjaya
Data ID2 vendor yang masih blank.xlsx
26 KB
Dear Pak Rozak,
Berikut ini informasi status PK/NP dari 351 vendor yang data ID2 dan data tax di SAP-nya masih blank.
Best Regards,
Shita Astari
Supplier Mgt. Team - PT SMART Tbk.

Gambar 3.66 Email user dengan lampiran data ID2 vendor dari SAP

Untuk mengatasi permasalahan ini, tim Finance mengirimkan file Excel tambahan yang berisi informasi vendor dengan status ID2 dan pajak yang telah diperbarui dari sistem SAP.



Gambar 3.67 Memasukan data ID2 vendor dari SAP kedalam PowerBI

```
tax_num_NonBlank =
IF(
    ref_vendor_MDG[tax_num2] = BLANK(),
    LOOKUPVALUE(ref_Vendor_Tax_Blank[ID2], ref_Vendor_Tax_Blank[Noven], ref_vendor_MDG[vendor
    ref_vendor_MDG[tax_num2]
)
```

Gambar 3.68 Kode DAX tax_num_NonBlank

Intern kemudian memanfaatkan data referensi ini (tabel ref_Vendor_Tax_Blank) dan menggabungkannya dengan data utama vendor (ref_vendor_MDG) menggunakan DAX bernama tax_num_NonBlank. DAX ini berfungsi untuk melengkapi data kosong dengan mencocokkan vendor berdasarkan kolom *Noven* dan mengambil nilai yang sesuai dari tabel referensi.

```
Vendor Tax =
VAR _Vendor_ID =
    CALCULATE(FIRSTNONBLANK(ref_vendor_MDG[tax_num_NonBlank],1),
    FILTER(ALL(ref_vendor_MDG),
    ref_vendor_MDG[country] = "ID"
    && PO[vendor_id] = ref_vendor_MDG[vendor]
    ))

VAR _Vendor_non_ID =
    CALCULATE(FIRSTNONBLANK(ref_vendor_MDG[tax_num_NonBlank],1),
    FILTER(ALL(ref_vendor_MDG),
    ref_vendor_MDG[country] <> "ID"
    && PO[vendor_id] = ref_vendor_MDG[vendor]
    ))

RETURN

IF(PO[scenario3] <> "Overseas" && _Vendor_ID<> BLANK(),_Vendor_ID,
    IF(_Vendor_non_ID<>BLANK(),_Vendor_non_ID))
```

Gambar 3.69 Kode DAX vendor Tax

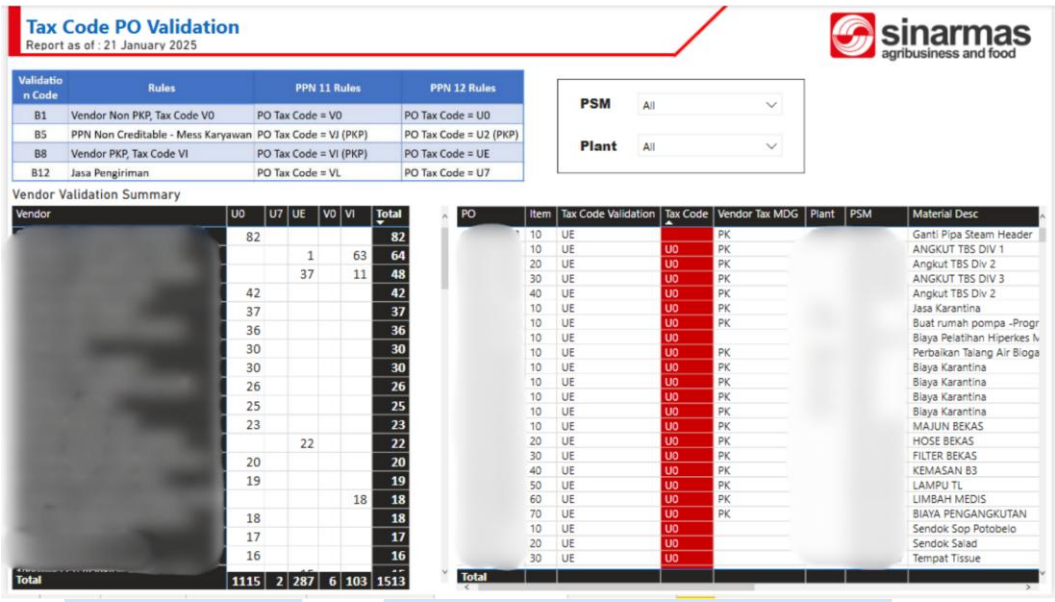
Selanjutnya, nilai yang telah diperbaiki tersebut digunakan dalam DAX Vendor Tax, yang menggabungkan informasi vendor berdasarkan negara asalnya (domestik atau non-domestik), sehingga setiap baris transaksi PO dapat menampilkan status pajak vendor yang valid. Dengan penerapan logika ini, dashboard dapat memberikan informasi yang lebih akurat dan mendukung kepatuhan terhadap regulasi perpajakan yang berlaku.

```
1  B14 =
2  IF(PO[po_creation_date] >= DATEVALUE("2025-01-12")
3      && PO[Vendor Tax] in {"PK","PKP"}
4      && NOT PO[tax_code] IN {"U6"}
5      && PO[material_desc] = "BERAS","B14",
6  IF(PO[po_creation_date] >= DATEVALUE("2025-01-12")
7      && PO[Vendor Tax] in {"NP"}
8      && NOT PO[tax_code] IN {"U0"}
9      && PO[material_desc] = "BERAS","B14",
10 IF(PO[po_creation_date] < DATEVALUE("2025-01-12")
11     && PO[Vendor Tax] in {"NP","PK","PKP"}
12     && NOT PO[tax_code] IN {"V0"}
13     && PO[material_desc] = "BERAS","B14")))
```

Gambar 3.70 Kode DAX validasi tax B14

Intern juga menambahkan validasi tax code baru khusus untuk material beras dengan kode validasi B14. Validasi ini dibuat untuk memastikan bahwa setiap purchase order yang berkaitan dengan material beras menggunakan tax code yang sesuai berdasarkan tanggal pembuatan PO, status pajak vendor, dan aturan yang berlaku. Secara rinci, validasi ini mengatur bahwa untuk PO yang dibuat setelah atau sama dengan 12 Januari 2025, jika vendor berstatus PK atau PKP, maka tax code yang digunakan tidak boleh termasuk kode tertentu seperti "U6". Sedangkan untuk vendor dengan status NP, tax code yang digunakan tidak boleh termasuk kode "U0". Selain itu, untuk PO yang dibuat sebelum tanggal tersebut, tax code yang dipakai juga harus sesuai dengan ketentuan lama, tidak boleh menggunakan kode "V0" untuk semua status pajak. Dengan penambahan validasi ini, intern memastikan bahwa penggunaan tax code untuk material beras sudah sesuai dengan peraturan yang berlaku dan dapat meminimalkan kesalahan dalam pencatatan pajak di dalam sistem.

3.2.2.13. Meningkatkan dan memperbaiki FA R2R PO Local Validation Monitoring Dashboard



Gambar 3.71 Tampilan dashboard FA R2R PO Local Validation Monitoring

Tugas lanjutan yang dilakukan oleh intern adalah merancang dan membangun sebuah dashboard baru dengan judul CBS-FA-R2R-PO_Local_Validation_Monitoring. Dashboard ini memiliki peran penting dalam mendukung kepatuhan perusahaan terhadap kebijakan perpajakan terbaru yang diberlakukan pemerintah sejak 12 Januari 2025, khususnya dalam proses pembelian lokal (*local procurement*) yang menggunakan sistem Purchase Order (PO) [11]. Perubahan regulasi ini mewajibkan perusahaan untuk melakukan penyesuaian terhadap penggunaan tax code berdasarkan status vendor (kena pajak atau tidak kena pajak) dan jenis transaksi yang dilakukan, agar pelaporan dan pembayaran pajak dapat dilakukan secara tepat dan sesuai ketentuan yang berlaku.

Validation Code	Rules	PPN 11 Rules	PPN 12 Rules
B1	Vendor Non PKP, Tax Code V0	PO Tax Code = V0	PO Tax Code = U0
B5	PPN Non Creditable - Mess Karyawan	PO Tax Code = VJ (PKP)	PO Tax Code = U2 (PKP)
B8	Vendor PKP, Tax Code VI	PO Tax Code = VI (PKP)	PO Tax Code = UE
B12	Jasa Pengiriman	PO Tax Code = VL	PO Tax Code = U7

Gambar 3.72 Kode validasi perajakan baru dan perpajakan lama

Dalam logika validasi ini, PO yang dibuat pada atau setelah 12 Januari 2025 untuk vendor kena pajak harus menggunakan tax code UE, sedangkan untuk vendor

tidak kena pajak, harus menggunakan tax code U0. Sementara itu, untuk PO sebelum 12 Januari 2025, vendor kena pajak harus menggunakan tax code VI, dan vendor tidak kena pajak menggunakan V0. Selain itu, terdapat ketentuan khusus untuk jenis transaksi tertentu: tax code U2 dan VJ digunakan untuk transaksi *mess karyawan*, sedangkan tax code U7 dan VL digunakan untuk transaksi jasa pengiriman.

Permasalahan utama yang dihadapi adalah banyaknya PO yang belum menggunakan tax code sesuai peraturan baru. Oleh karena itu, intern bertugas mengembangkan sejumlah kode validasi (B1, B5, B8, B12, dan B14) dalam bentuk formula DAX, untuk mengecek kesesuaian tax code terhadap data PO, seperti tanggal pembuatan, status PKP vendor, dan konten dari field. Berikut penjelasan tiap validasi.

```
1  B1 =
2  VAR IsPKP = PO_Local[Vendor Tax MDG] IN {"PK", "PKP"} || SEARCH(" PT", PO_Local[name_of_vendor], 1, 0)
3  VAR IsNonPKP = NOT IsPKP
4  VAR IsAfterJan12 = PO_Local[po_creation_date] >= DATE(2025, 1, 12)
5
6  -- Exclude Perumahan Staff
7  VAR IsPerumahanStaff =
8      CONTAINSSTRING(PO_Local[short_text], "R-Staff") ||
9      (CONTAINSSTRING(PO_Local[short_text], "perumahan") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
10     (CONTAINSSTRING(PO_Local[short_text], "rumah") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
11     (CONTAINSSTRING(PO_Local[short_text], "perum") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
12     (CONTAINSSTRING(PO_Local[short_text], "RMH") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
13     (CONTAINSSTRING(PO_Local[short_text], "Perunahan") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
14     CONTAINSSTRING(PO_Local[short_text], "R.Staff") ||
15     (CONTAINSSTRING(PO_Local[short_text], "RUMDIS") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
16     CONTAINSSTRING(PO_Local[short_text], "R Staff")
17
18 -- Exclude Jasa Pengiriman
19 VAR IsShippingService =
20     (CONTAINSSTRING(PO_Local[short_text], "pengiriman")
21     || CONTAINSSTRING(PO_Local[short_text], "kirim")
22     || CONTAINSSTRING(PO_Local[short_text], "bill of lading")) &&
23     NOT CONTAINSSTRING(PO_Local[short_text], "brondolan") &&
24     NOT CONTAINSSTRING(PO_Local[short_text], "buah") &&
25     NOT CONTAINSSTRING(PO_Local[short_text], "sawit") &&
26     NOT CONTAINSSTRING(PO_Local[short_text], "tbs") &&
27     NOT CONTAINSSTRING(PO_Local[short_text], "limbah") &&
28     NOT CONTAINSSTRING(PO_Local[short_text], "asuransi")
29
30 VAR ShouldExclude = IsPerumahanStaff || IsShippingService
31
32 VAR Result =
33     IF(
34         IsNonPKP && NOT ShouldExclude,
35         IF(IsAfterJan12, "U0","V0"))
36 RETURN Result
```

Gambar 3.73 Kode DAX validasi tax B1

Validasi B1 dilakukan untuk mengecek penggunaan tax code V0 dan U0 pada Purchase Order (PO) yang dibuat oleh vendor non-PKP dan tidak berkaitan dengan

transaksi perumahan staff atau jasa pengiriman. Validasi ini mempertimbangkan tanggal pembuatan PO sebagai acuan utama. Jika PO dibuat setelah tanggal 12 Januari 2025, maka tax code yang seharusnya digunakan adalah U0. Sebaliknya, jika PO dibuat sebelum tanggal tersebut, maka tax code yang sesuai adalah V0.

```
B5 =
VAR IsAfterJan12 = PO_Local[po_creation_date] >= DATE(2025, 1, 12)

VAR IsStaffHousing =
    CONTAINSSTRING(PO_Local[short_text], "R-Staff") ||
    (CONTAINSSTRING(PO_Local[short_text], "perumahan") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    (CONTAINSSTRING(PO_Local[short_text], "rumah") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    (CONTAINSSTRING(PO_Local[short_text], "perum") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    (CONTAINSSTRING(PO_Local[short_text], "RMH") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    (CONTAINSSTRING(PO_Local[short_text], "Perunahan") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    CONTAINSSTRING(PO_Local[short_text], "R.Staff") ||
    (CONTAINSSTRING(PO_Local[short_text], "RUMDIS") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    CONTAINSSTRING(PO_Local[short_text], "R Staff")

VAR IsPKP = PO_Local[Vendor Tax MDG] IN {"PK", "PKP"}

VAR Result =
    IF(IsStaffHousing,
        IF(IsAfterJan12,
            IF(IsPKP, "U2", "U0"),
            IF(IsPKP, "VJ", "V0")))

RETURN Result
```

Gambar 3.74 Kode DAX validasi tax B5

Validasi B5 berfokus pada transaksi terkait perumahan staff. PO yang mengandung kata kunci seperti “rumah”, “sewa rumah”, “penginapan”, dan sejenisnya akan diperiksa lebih lanjut berdasarkan status PKP vendor dan tanggal PO. Jika vendor termasuk PKP dan PO dibuat setelah 12 Januari 2025, maka tax code yang benar adalah U2. Jika PO dibuat sebelum tanggal tersebut, maka harus menggunakan VJ. Untuk vendor non-PKP, aturan ini tidak berlaku dan akan dikembalikan ke logika B1 (V0 atau U0).

```
B8 =
VAR IsPKP = PO_Local[Vendor Tax MDG] IN {"PK", "PKP"} || SEARCH(" PT", PO_Local[name_of_vendor], 1, 0) > 0
VAR IsAfterJan12 = PO_Local[po_creation_date] >= DATE(2025, 1, 12)

-- Exclude Perumahan Staff
VAR IsPerumahanStaff =
    CONTAINSSTRING(PO_Local[short_text], "R-Staff") ||
    (CONTAINSSTRING(PO_Local[short_text], "perumahan") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    (CONTAINSSTRING(PO_Local[short_text], "rumah") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    (CONTAINSSTRING(PO_Local[short_text], "perum") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    (CONTAINSSTRING(PO_Local[short_text], "RMH") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    (CONTAINSSTRING(PO_Local[short_text], "Perunahan") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    CONTAINSSTRING(PO_Local[short_text], "R.Staff") ||
    (CONTAINSSTRING(PO_Local[short_text], "RUMDIS") && CONTAINSSTRING(PO_Local[short_text], "staff")) ||
    CONTAINSSTRING(PO_Local[short_text], "R Staff")

-- Exclude Jasa Pengiriman
VAR IsShippingService =
    (CONTAINSSTRING(PO_Local[short_text], "pengiriman")
    || CONTAINSSTRING(PO_Local[short_text], "kirim")
    || CONTAINSSTRING(PO_Local[short_text], "bill of lading")) &&
    NOT CONTAINSSTRING(PO_Local[short_text], "brondolan") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "buah") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "sawit") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "tbs") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "limbah") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "asuransi")


VAR ShouldExclude = IsPerumahanStaff || IsShippingService

VAR Result =
    IF(IsPKP && NOT ShouldExclude,
        IF( IsAfterJan12, "UE", "VI"))

RETURN Result
```

Gambar 3.75 Kode DAX validasi tax B8

Validasi B8 digunakan untuk memverifikasi bahwa PO dari vendor PKP yang tidak termasuk dalam kategori perumahan staff atau jasa pengiriman telah menggunakan tax code yang sesuai. Jika PO dibuat setelah 12 Januari 2025, tax code yang seharusnya digunakan adalah UE, sedangkan jika dibuat sebelum tanggal tersebut, maka yang digunakan adalah VI. Validasi ini membantu memastikan bahwa transaksi dari vendor PKP yang bersifat umum mengikuti ketentuan perpajakan yang terbaru, tanpa tercampur dengan pengecualian seperti perumahan atau logistik.

```
B12 =
VAR IsShippingService =
    (CONTAINSSTRING(PO_Local[short_text], "pengiriman")
    || CONTAINSSTRING(PO_Local[short_text], "kirim")
    || CONTAINSSTRING(PO_Local[short_text], "bill of lading")) &&
    NOT CONTAINSSTRING(PO_Local[short_text], "brondolan") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "buah") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "sawit") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "tbs") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "limbah") &&
    NOT CONTAINSSTRING(PO_Local[short_text], "asuransi")


VAR IsAfterJan12 = PO_Local[po_creation_date] >= DATE(2025, 1, 12)

VAR Result =
    IF(IsShippingService,
        IF(IsAfterJan12, "U7", "VL"))

RETURN Result
```

Gambar 3.76 Kode DAX validasi tax B12

Terakhir adalah Validasi B12 dirancang khusus untuk mengidentifikasi transaksi yang berkaitan dengan jasa pengiriman, namun tidak melibatkan pengiriman buah, limbah, atau sawit. PO yang mengandung kata kunci seperti “jasa pengiriman”, “ekspedisi”, atau “angkut” akan diperiksa. Jika tanggal PO lebih dari 12 Januari 2025, maka tax code yang benar adalah U7. Jika PO dibuat sebelum tanggal tersebut, maka tax code yang sesuai adalah VL. Validasi ini mengakomodasi perlakuan perpajakan yang berbeda terhadap jasa logistik setelah kebijakan baru diberlakukan.

```
Tax Code Validation =
VAR Validation = [B1] & [B8] & [B12] & [B5]
RETURN
IF(
    Validation = PO_Local[tax_code],
    BLANK(),
    Validation
)
```

Gambar 3.77 Kode DAX Tax Code Validation

Setelah seluruh logika validasi pajak (B1, B5, B8, dan B12) selesai dibuat, langkah selanjutnya adalah menggabungkannya ke dalam satu kolom bernama Tax Code Validation. Kolom ini berfungsi untuk membandingkan hasil validasi dengan kode pajak aktual pada data PO. Penggabungan dilakukan dengan memprioritaskan validasi tertentu menggunakan fungsi COALESCE, yaitu memeriksa B5, kemudian B12, B1, dan terakhir B8. Jika hasil validasi sesuai dengan kode pajak yang tercatat, maka kolom akan dibiarkan kosong. Namun, jika terdapat perbedaan, maka akan ditampilkan kode pajak yang seharusnya. Selanjutnya, nilai kode pajak tercatat yang tidak sesuai akan otomatis ditandai dengan warna merah di dashboard, sehingga memudahkan tim dalam mengidentifikasi dan memperbaiki kesalahan entri pajak.

```
Tax Code Validation Non Blank =
[B1] & [B8] & [B12] & [B5]
```

Gambar 3.78 Kode DAX Tax Code Validation Non Blank

Setelah kolom Tax Code Validation berhasil dibuat untuk membandingkan hasil validasi dengan kode pajak aktual, ditemukan bahwa kolom ini hanya menampilkan kode pajak jika terdapat ketidaksesuaian atau butuh validasi. Untuk keperluan analisis yang lebih menyeluruh, intern kemudian membuat kolom tambahan bernama Tax Code Validation Non Blank yang menggabungkan seluruh hasil validasi (B1, B5, B8, dan B12) tanpa mempertimbangkan kesesuaian, sehingga seluruh nilai kode pajak (baik yang benar maupun yang perlu revisi) tetap ditampilkan.

```
Tax Code Revision =
IF(
    ISBLANK(PO_Local[Tax Code Validation Non Blank]),
    BLANK(),
    IF(
        PO_Local[tax_code] <> PO_Local[Tax Code Validation Non Blank],
        "Tax Code Revision",
        BLANK()
    )
)
```

Gambar 3.79 Kode DAX Tax Code Revision

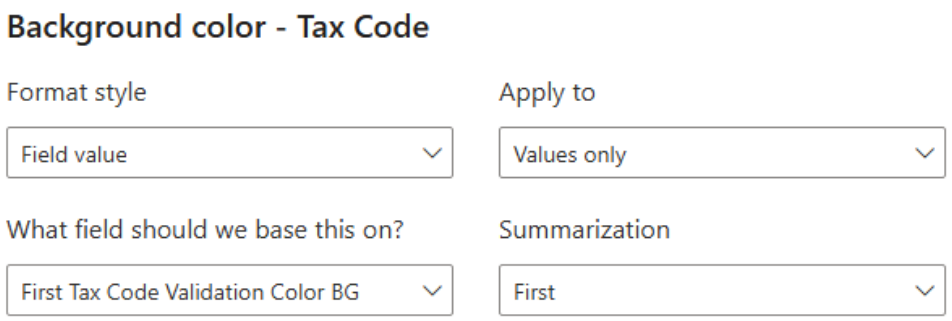
Untuk mempermudah proses identifikasi entri data yang memerlukan validasi ulang, intern membuat kolom tambahan bernama Tax Code Revision. Kolom ini berfungsi sebagai indikator teks yang secara otomatis menampilkan label *"Tax Code Revision"* pada setiap baris data yang memiliki perbedaan antara kode pajak

aktual dan hasil validasi gabungan (dari B1, B5, B8, dan B12). Dengan menggunakan fungsi logika DAX, hanya baris yang memiliki selisih antara data aktual dan hasil validasi yang akan dilabeli, sementara baris lain yang sudah sesuai dibiarkan kosong (blank). Kode ini berguna untuk membantu mempercepat proses pengecekan data.

```
1 Tax Code Validation Color BG =
2 IF(NOT PO_Local[Tax Code Revision] = BLANK(),"#cc0000",BLANK())
```

Gambar 3.80 Kode DAX Tax Code Validation Color BG

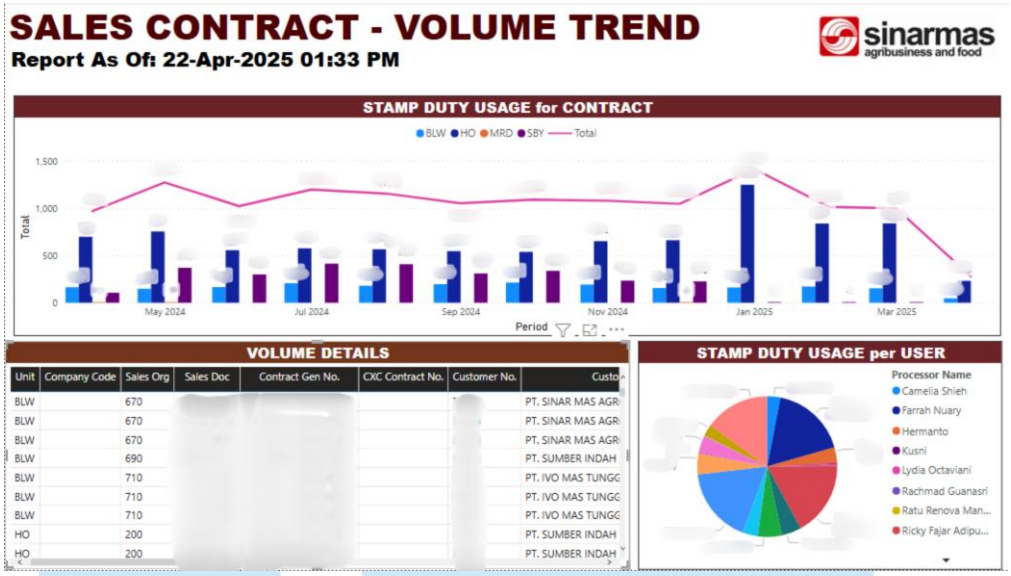
Untuk mendukung keterbacaan visual yang lebih baik dalam dashboard, intern juga menambahkan kolom Tax Code Validation Color BG. Kolom ini digunakan untuk memberikan pewarnaan latar belakang pada baris-baris yang telah diberi label *Tax Code Revision*. Apabila suatu baris ditandai sebagai membutuhkan revisi, maka latar belakang kolom kode pajak akan diberi warna merah (#cc0000).



Gambar 3.81 Conditional formatting dengan field value Tax Code Validation Color BG

Sebagai langkah akhir, kolom Tax Code Validation Color BG dimasukkan ke dalam fitur *Conditional Formatting* pada visual tabel di dashboard Power BI. Tujuannya adalah agar warna latar belakang yang telah ditentukan—dalam hal ini merah untuk entri yang memerlukan revisi kode pajak—dapat secara otomatis diterapkan pada baris yang relevan. Dengan pengaturan ini, pengguna dapat dengan mudah melihat dan membedakan baris yang mengandung potensi kesalahan tanpa perlu membaca satu per satu.

3.2.2.14. Membuat dashboard untuk Stamp Duty Report Finance dan Procurement



Gambar 3.82 Tampilan dashboard Stamp Duty Report Finance dan Procurement

Dashboard Stamp Duty Report dibuat untuk membantu proses monitoring dan pelaporan penggunaan materai (stamp duty) pada dokumen transaksi di atas 10 juta rupiah. Dashboard ini digunakan oleh dua tim utama, yaitu tim Sales dan tim Purchase. Tim Sales menggunakan materai untuk dokumen kontrak dan billing, sedangkan tim Purchase menggunakannya untuk dokumen pembelian. Dengan adanya dashboard ini, pihak perusahaan dapat memastikan bahwa pemakaian materai dilakukan sesuai ketentuan yang berlaku dan tidak terjadi kelebihan penggunaan yang bisa berdampak pada audit maupun beban biaya tambahan.

Issue yang dialami oleh user pada dashboard *Stamp Duty Report* adalah terkait dengan pelaporan penggunaan materai untuk customer tertentu yang proses pelaporannya dilakukan dalam bentuk *Invoice List (ZLR)*. Dalam kasus ini, satu Assignment dapat memiliki beberapa *Billing Document* yang seluruhnya diawali dengan angka "009". Berdasarkan logic DAX yang telah ada sebelumnya, setiap *Billing Document* tersebut secara otomatis dihitung menggunakan satu materai, sehingga menyebabkan overcount atau kelebihan dalam perhitungan total stamp duty.Namun pada praktik sebenarnya, untuk setiap *Invoice List (ZLR)* hanya diperlukan satu materai saja, yang mewakili seluruh *Billing Document* dalam satu

Assignment tersebut. Dengan kata lain, hanya satu dokumen—yaitu *Billing Document* dengan nomor terbesar—yang perlu dilaporkan sebagai representasi penggunaan satu stamp

```
Exception Invoice List =
var b = CALCULATE(FIRSTNONBLANK('Billing Exception List'[Customer Name],0),
FILTER(
    ALL('Billing Exception List'), 'Billing Exception List'[Customer Code] = ZFBL5N[Customer]
    && 'Billing Exception List'[Remarks] = "Invoice List (ZLR)"))
return
IF(b <> BLANK() && LEFT(ZFBL5N[Billing_Document],3) = "009", "Invoice List")
```

Gambar 3.83 Kode DAX Exeption Invoice List

Untuk mengidentifikasi Billing Document mana saja yang termasuk dalam kategori Invoice List, dibuatlah DAX bernama Exception Invoice List. DAX ini memeriksa apakah Customer tersebut tercatat dalam tabel pengecualian Billing Exception List dengan keterangan *Invoice List (ZLR)*, dan memastikan dokumen tersebut memiliki prefix "009". Bila kondisi ini terpenuhi, maka dokumen akan ditandai sebagai "Invoice List" dan akan diperlakukan secara khusus dalam proses penghitungan.

```
Value Invoicelist =
var b = LOOKUPVALUE('Billing Exception List'[Qty],'Billing Exception List'[Customer Code],'ZFBL5N'[Customer])
return
if(b <> BLANK(),
    VAR CurrentAssignment = ZFBL5N[Assignment]
    VAR CurrentBilling = ZFBL5N[Billing_Document]
    VAR MaxBilling =
        CALCULATE(
            MAX(ZFBL5N[Billing_Document]),
            FILTER(
                ZFBL5N,
                LEFT(ZFBL5N[Billing_Document], 3) = "009"
                    && ZFBL5N[Assignment] = CurrentAssignment
            )
        )
    RETURN
        IF(CurrentBilling = MaxBilling && B=1 ,1,0),0)
```

Gambar 3.84 Kode DAX Value Invoicelist

Selanjutnya, untuk mencegah terjadinya penghitungan ganda terhadap penggunaan materai pada *Invoice List*, digunakan rumus DAX tambahan bernama Value Invoicelist. Rumus ini mengecek apakah customer termasuk dalam daftar pengecualian, lalu mencari *Billing Document* dengan nomor terbesar dalam satu *Assignment*. Hanya *Billing Document* terakhir (nomor terbesar) yang dihitung sebagai satu penggunaan materai, selama jumlah yang terdaftar dalam pengecualian

adalah satu (*Qty = 1*). Dengan logika ini, penghitungan menjadi lebih akurat dan mencerminkan kondisi riil di lapangan, terutama dalam pelaporan dokumen berharga tinggi. Langkah terakhir adalah mengaktifkan filter pada kolom Exception Invoice List dengan nilai *Invoice List*, serta kolom Value Invoicelist dengan nilai *1*, untuk memastikan hanya dokumen yang valid yang dihitung dalam total penggunaan stamp.

3.2.3 Pembuatan SQL untuk Pengambilan Email Vendor Terkait TAX/Finance (UNION ALL)

Dalam tugas database ini, Intern diminta untuk membuat query SQL guna menentukan alamat email yang paling tepat untuk digunakan dalam komunikasi dengan vendor. Kebutuhan utamanya adalah memilih email vendor berdasarkan prioritas, yaitu jika terdapat email dengan catatan "tax" atau "finance", maka email tersebut harus diutamakan. Jika tidak ada, maka digunakan email lainnya seperti dari kategori "proc" atau umum. Setiap vendor dapat memiliki beberapa alamat email dengan berbagai catatan (email notes), sehingga perlu dilakukan pengelompokan dan seleksi berdasarkan tingkat prioritas.

SQLQuery1.sql - 10....BS_Fin_Report (77))*

```
WITH Ranked_Vendors AS (
    SELECT
        vendor_email.vendor_number,
        vendor_name.name AS vendor_name,
        vendor_email.email_address,
        ROW_NUMBER() OVER (PARTITION BY vendor_email.vendor_number ORDER BY vendor_email.priority DESC) AS rnk
    FROM (SELECT
            vendor_number,
            concat_ws('; ', collect_list(email_address)) AS email_address,
            1 AS priority -- Priority for TAX or finance emails
        FROM (SELECT DISTINCT
                vendor_number,
                lower(email_address) AS email_address
            FROM (SELECT DISTINCT
                    vendor_number,
                    email_notes,
                    lower(email_address) AS email_address,
                    row_number() OVER (PARTITION BY vendor_number ORDER BY email_notes DESC) AS ranking
                FROM mgmt_coe_db.vw_pbi_dm_vendor_email
                WHERE
                    lower(email_address) NOT RLIKE '@sinarmas-agri|vendor|dummy|c|_'
                    AND vendor_number NOT RLIKE 'E'
                    AND (email_notes = 'TAX' OR LOWER(email_notes) LIKE '%finance%')
            ) rank_vendor
            WHERE ranking <= 20
        ) vendor
        GROUP BY vendor_number
```

Gambar 3.85 Query SQL untuk Mengambil Data Vendor (bagian 1)

```
        UNION ALL
        SELECT
            vendor_number,
            concat_ws(', ', collect_list(email_address)) AS email_address,
            0 AS priority -- Lower priority for non-TAX emails
        FROM (SELECT DISTINCT
                vendor_number,
                lower(email_address) AS email_address
            FROM (SELECT DISTINCT
                    vendor_number,
                    email_notes,
                    lower(email_address) AS email_address,
                    row_number() OVER (PARTITION BY vendor_number ORDER BY email_notes DESC) AS ranking
                FROM mgmt_coe_db.vw_pbi_dm_vendor_email
                WHERE
                    lower(email_address) NOT RLIKE '@sinarmas-agri|vendor|dummy|<| '
                    AND vendor_number NOT RLIKE 'E'
                    AND (email_notes != 'TAX' AND LOWER(email_notes) NOT LIKE '%finance%')
            ) rank_vendor
            WHERE ranking <= 20
        ) vendor
        GROUP BY vendor_number
    ) vendor_email
    JOIN (SELECT DISTINCT
            vendor,
            name
        FROM mgmt_coe_db.vw_pbi_dm_vendor_all
    ) vendor_name
    ON vendor_email.vendor_number = vendor_name.vendor
)
SELECT vendor_number, vendor_name, email_address
FROM Ranked_Vendors
WHERE rnk = 1
ORDER BY vendor_number;
```

Gambar 3.86 **Query SQL untuk Mengambil Data Vendor (bagian 2)**

Untuk menjawab kebutuhan tersebut, Intern mengembangkan query SQL yang terdiri dari beberapa langkah. Pertama, query memfilter data email yang valid dan mengeliminasi alamat email yang bersifat dummy atau sistem. Kemudian, Intern memberi prioritas pada email dengan catatan "tax" atau "finance". Selanjutnya, query menggabungkan email-email dengan prioritas tertinggi per vendor menjadi satu baris agar mudah dibaca dan digunakan. Jika email dengan prioritas tinggi tidak tersedia, maka email lainnya akan diambil sebagai alternatif. Hasil akhirnya adalah daftar vendor beserta alamat email yang paling relevan, digabung jika lebih dari satu, dan siap digunakan untuk keperluan komunikasi internal perusahaan.

3.3 Kendala yang Ditemukan

Beberapa kendala yang ditemukan intern saat melaksanakan proses magang pada perusahaan Sinarmas Agribusiness and Food adalah:

a) Penugasan di Tim SAP yang Tidak Sesuai dengan Minat dan Penjurusan

Pada awal magang, intern ditempatkan di tim SAP yang fokus pada transformasi sistem ERP perusahaan. Penugasan ini kurang sesuai dengan latar belakang dan

minat intern yang lebih condong ke bidang analisis data dan pengembangan dashboard, sehingga menyebabkan penurunan motivasi di tahap awal.

b) Perangkat Kerja yang Kurang Optimal

Pada awal masa magang, intern sempat diberikan laptop pribadi untuk menunjang kegiatan kerja. Namun, setelah beberapa hari, laptop tersebut dialihkan untuk digunakan oleh karyawan baru. Akibatnya, intern harus bekerja menggunakan komputer kantor bersama yang memiliki spesifikasi rendah dan kinerja yang lambat. Hal ini menyulitkan intern dalam menjalankan aplikasi berat seperti Power BI, serta menghambat proses analisis data dan pembuatan dashboard secara efisien.

c) Gangguan Server Kantor

Selama kurang lebih dua minggu, intern menghadapi kendala besar akibat server kantor yang mengalami gangguan. Akibatnya, intern tidak dapat mengakses folder bersama yang berisi data penting untuk keperluan pengerjaan proyek, sehingga proses kerja menjadi terhenti sementara.

d) Ketidaktahuan terhadap Istilah-istilah Teknis di Kantor

Intern memerlukan waktu untuk beradaptasi dengan istilah teknis yang umum digunakan di lingkungan kerja, seperti *Nomat* (Nomor Material), *SO* (Sales Order), *SQ* (Sales Quotation), *FG* (Finished Goods), *RFQ* (Request for Quotation), *MRP* (Material Requirement Planning), *MPS* (Master Production Schedule), dan lain-lain. Ketidaktahuan awal terhadap istilah ini sempat menyebabkan kebingungan dalam memahami instruksi dan dokumen pekerjaan.

e) Penggunaan Power BI yang Lebih Kompleks Dibandingkan di Kampus

Meskipun intern telah mempelajari dasar Power BI di kampus, namun penerapan Power BI di lingkungan kerja jauh lebih kompleks. Intern harus memahami berbagai sumber data, penggunaan DAX lanjutan, serta penggabungan data antar tabel yang rumit.

f) Kendala dalam Uji Validitas Data di Power BI

Salah satu kesulitan teknis yang dihadapi intern adalah dalam memvalidasi data di Power BI. Terkadang, formula DAX yang disusun tidak menghasilkan output sesuai harapan. Masalah ini sering kali merupakan akibat dari kompleksitas model data

dan ketidaksesuaian antara data aktual dengan asumsi yang digunakan dalam perhitungan.

3.4 Solusi atas Kendala yang Ditemukan

Solusi atas kendala yang ditemukan selama proses kerja magang pada perusahaan Sinarmas Agribusiness and Food adalah:

a) Penugasan di Tim SAP yang Tidak Sesuai dengan Minat dan Penjurusan

Intern mengajukan permintaan kepada supervisor untuk dipindahkan dari tim SAP ke proyek yang lebih sesuai, yaitu pengembangan dashboard, agar dapat bekerja sesuai dengan minat dan jurusan Big Data.

b) Perangkat Kerja yang Kurang Optimal

Intern tetap menggunakan komputer kantor meskipun memiliki spesifikasi rendah, dan menyiasatinya dengan menyeleksi kolom dan melakukan data cutting agar proses analisis dan visualisasi data berjalan lebih ringan.

c) Gangguan Server Kantor

Saat server kantor mengalami gangguan, intern mengalihkan fokus ke pekerjaan yang tidak memerlukan koneksi server seperti dokumentasi proses kerja dan menyusun formula DAX secara offline.

d) Ketidaktahuan terhadap Istilah-istilah Teknis di Kantor

Intern mempelajari istilah-istilah teknis yang digunakan di lingkungan kerja dengan aktif bertanya kepada supervisor dan staf, serta mencatat istilah baru untuk mempercepat pemahaman terhadap proses bisnis.

e) Penggunaan Power BI yang Lebih Kompleks Dibandingkan di Kampus

Intern memperdalam kemampuan Power BI dengan belajar langsung dari staf kantor dan menambah pengetahuan secara mandiri melalui tutorial-tutorial di YouTube.

f) Kendala dalam Uji Validitas Data di Power BI

Ketika terjadi kesalahan dalam validasi data di Power BI, intern melakukan evaluasi ulang terhadap struktur data dan menyusun kembali rumus DAX untuk mendapatkan hasil yang sesuai dengan kebutuhan analisis.