BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi Magang

Selama pelaksanaan program magang di Sinar Mas Land, dijalankan peran sebagai *Data Governance* Intern yang berada di bawah Divisi *Master Data Management* (MDM). Dalam struktur organisasi, posisi ini berada di bawah supervisi langsung *Data Governance* Staff yang selanjutnya melaporkan kegiatan serta hasil kerja kepada *Master Data Management* Section Head. Peran tersebut ditempatkan sebagai bagian dari tim yang bertanggung jawab dalam menjaga kualitas, konsistensi, dan integritas data pada sistem perusahaan, khususnya pada dua sistem utama yang digunakan, yaitu *SAP Customer Experience* (CX) dan SAP S/4HANA.

Fokus utama diarahkan pada pengelolaan serta validasi data yang bersumber dari SAP CX, sementara intern lainnya menangani data yang berasal dari SAP S/4HANA. Pembagian tanggung jawab ini diterapkan agar proses pemantauan kualitas data dapat dilakukan secara lebih terfokus dan berjalan secara efisien.

Dalam pelaksanaan aktivitas harian, dilakukan koordinasi secara intensif dengan *Data Governance* Staff yang memberikan arahan teknis sekaligus penjelasan terkait konteks data yang sedang dikerjakan. Koordinasi tersebut mencakup pemahaman alur kerja, penerapan standar operasional prosedur (SOP) dalam proses validasi data, serta penentuan skala prioritas pekerjaan harian. Selain itu, dilakukan komunikasi secara berkala dengan Section Head untuk memastikan bahwa hasil pekerjaan telah sesuai dengan standar kualitas yang ditetapkan oleh tim MDM. Pola komunikasi yang diterapkan bersifat terbuka, sehingga dapat berdiskusi maupun meminta klarifikasi apabila diperlukan, terutama ketika menemukan anomali data atau kasus yang memerlukan analisis lebih lanjut.

Bentuk koordinasi yang dilakukan mencakup berbagai kegiatan teknis, seperti penyusunan dan eksekusi *query* SQL untuk pengecekan data, identifikasi data yang tidak valid, deteksi duplikasi, hingga analisis konsistensi data antar sistem antara CX dan S/4. Dalam kegiatan magang ini, dilibatkan juga dalam proses pemantauan rutin terhadap beberapa dataset utama yang menjadi fokus tim *Data Governance*, antara lain data customer individual, customer corporate, serta data business partner pada sistem S4. Pada kondisi tertentu, koordinasi turut dilakukan bersama intern lain untuk membahas temuan data yang saling berkaitan antara kedua sistem. Kolaborasi tersebut diperlukan agar hasil validasi data dapat mencerminkan kondisi data yang aktual dan tetap konsisten lintas platform.

Pelaksanaan kerja secara *onsite* di kantor Sinar Mas Land memberikan kemudahan tersendiri dalam proses koordinasi. Interaksi dapat dilakukan secara langsung tanpa harus melalui mekanisme formal, seperti penjadwalan rapat, sehingga diskusi terkait permasalahan data dapat dilakukan dengan lebih cepat. Sehingga dapat diperoleh umpan balik secara langsung, melakukan penyesuaian terhadap hasil pekerjaan, serta menyelesaikan perbaikan data secara lebih efisien. Kondisi ini mendukung efektivitas proses kerja sekaligus membantu untuk memahami alur operasional tim MDM secara lebih menyeluruh.

3.2 Tugas dan Uraian Kerja Magang

Peserta magang yang ditempatkan pada bagian *Data Governance* memiliki tanggung jawab dalam menyusun serta mengeksekusi *query* sesuai dengan kebutuhan tim, sekaligus menjalankan daily task yang telah ditetapkan. Tugas-tugas yang diberikan tidak bersifat statis, melainkan berkembang secara bertahap dan menyesuaikan dengan dinamika kebutuhan operasional tim. Rincian dari pelaksanaan tugas tersebut kemudian dirangkum dan disajikan secara sistematis dalam Tabel 1.

Tabel 1 Uraian Kerja Magang

No.	Minggu	Aktivitas	Keterangan
1	September (Minggu 1–2)	*Onboarding* ke divisi *Data Governance*; pengenalan struktur perusahaan dan alur kerja; memahami pembagian	Tahap adaptasi awal

No.	Minggu	Aktivitas	Keterangan
		tugas CX dan S4; mempelajari SOP validasi data; peninjauan struktur tabel customer	
2	September (Minggu 2–3)	Mempelajari kategori error pada SAP CX (kapitalisasi nama, email invalid, nomor telepon, NPWP); membuat *query* dasar validasi; latihan mencari data duplikat; mulai membantu pengecekan KTP	Mulai menangani data teknis
3	September (Minggu 3–Sekarang)	Menjalankan tugas daily task	Mulai memegang rutinitas harian dan menjalankan *query* yang telah disusun menggunakan SQL Server Management Studio dan laporan via excel
4	October (Minggu 1–Sekarang)	Mulai menjalankan tugas berdasarkan permintaan	Mulai menerima tugas berdasarkan permintaan seperti permintaan pengecekan KTP kosong, Mobile kosong hingga melakukan enrichment pada data yang kosong tersebut

Gambar 3.1 Flowchart Proses Kerja Daily Task

Dalam pelaksanaan tugas harian sebagai intern pada tim *Data*

Governance, terdapat alur kerja yang dijalankan secara terstruktur dan berulang setiap hari untuk memastikan kualitas data customer tetap terjaga, sebagaimana ditunjukkan pada Gambar 3.1. Proses ini diawali dengan penyusunan serta penyesuaian *query* validasi. Meskipun *query* utama telah disiapkan sejak awal masa magang, penyesuaian tetap dilakukan setiap hari pada parameter waktu, khususnya pada bagian create date. Penyesuaian ini diperlukan karena proses pengecekan difokuskan pada data terbaru dengan periode H-1, sehingga tanggal pada *query* perlu diperbarui sebelum proses validasi dijalankan. Langkah tersebut bertujuan agar seluruh data customer yang masuk pada hari sebelumnya dapat diperiksa secara menyeluruh tanpa ada data yang terlewat.

Setelah parameter *query* diperbarui, *query* kemudian dijalankan untuk melakukan pengecekan kualitas data. Pada tahap ini, seluruh aturan validasi dieksekusi, yang mencakup pemeriksaan kelengkapan atribut, kesesuaian format penulisan, keakuratan data identitas, keunikan data untuk mencegah terjadinya duplikasi, serta konsistensi antar kolom yang saling berkaitan. Hasil dari proses ini berupa daftar data customer yang dikategorikan sebagai data tidak valid. Data tersebut selanjutnya dianalisis lebih lanjut untuk mengetahui penyebab kesalahan serta menentukan prioritas penanganan yang diperlukan.

Tahapan berikutnya adalah melakukan pemeriksaan secara rinci terhadap setiap data yang teridentifikasi tidak valid. Pada proses ini, dilakukan pengecekan terhadap create date untuk memastikan bahwa data tersebut termasuk dalam periode pemeriksaan harian. Selain itu, proses ini juga dimanfaatkan untuk mengidentifikasi kemungkinan adanya pola kesalahan yang berulang dari PIC atau unit tertentu. Verifikasi lanjutan dilakukan berdasarkan jenis error yang ditemukan, seperti ketidaksesuaian format penulisan nama dengan standar perusahaan, format email yang tidak valid, nomor telepon yang tidak konsisten, kesalahan pengisian NPWP, potensi duplikasi data customer, maupun ketidaksesuaian informasi identitas lainnya. Pemeriksaan secara rinci ini dilakukan agar data yang dikategorikan tidak valid benar-benar akurat dan siap untuk ditindaklanjuti.

Setelah proses validasi manual selesai dilakukan, seluruh data yang

teridentifikasi tidak valid disusun dalam bentuk laporan harian. Laporan ini memuat daftar data yang memerlukan perbaikan, kategori kesalahan yang ditemukan, serta informasi pendukung lain yang relevan. Penyusunan laporan dilakukan dengan format yang konsisten agar mudah dipahami oleh atasan maupun PIC yang bertanggung jawab. Laporan tersebut berfungsi sebagai alat pemantauan kualitas data sekaligus menjadi dasar tindak lanjut bagi pihak yang melakukan input data.

Tahap terakhir dalam alur kerja harian adalah penyampaian hasil kepada atasan. Setiap hari, laporan data tidak valid dikirimkan kepada atasan melalui email untuk selanjutnya diteruskan kepada Customer Service (CS) atau pihak terkait yang melakukan input data. Atasan kemudian menyampaikan laporan tersebut kepada PIC yang bersangkutan agar dapat dilakukan konfirmasi ulang kepada customer atau perbaikan data melalui sistem. Mekanisme ini memastikan bahwa setiap temuan data tidak valid dapat ditindaklanjuti melalui jalur koordinasi yang resmi serta membantu memastikan proses perbaikan berjalan secara efektif. Dalam konteks ini, tanggung jawab perbaikan data tetap berada pada pihak penginput, sementara tim *Data Governance* berperan sebagai pengawas dan pengendali kualitas data.

Melalui alur kerja harian yang terstruktur ini, proses *Monitoring* kualitas data dapat berjalan secara konsisten dan berkelanjutan. Proses yang sama diterapkan setiap hari untuk menjaga kualitas data customer, memastikan penerapan standar *Data Governance*, serta mendukung keandalan data yang digunakan dalam proses bisnis perusahaan. Selain itu, alur kerja ini turut memperkuat koordinasi antara tim *Data Governance*, atasan, dan unit kerja operasional yang terlibat dalam proses penginputan data.

Gambar 3.2 Flowchart Proses Kerja by Request

Pada Pada alur kerja task *by request* sebagaimana ditunjukkan pada Gambar 3.2, proses diawali ketika atasan atau pihak terkait menyampaikan kebutuhan khusus yang memerlukan penanganan data secara spesifik. Permintaan tersebut umumnya berkaitan dengan aktivitas seperti membandingkan data antara sistem MDM dan SAP CX untuk memastikan kesesuaian informasi, menelusuri perbedaan pada data identitas customer, maupun melakukan penarikan data dari database berdasarkan kriteria tertentu. Tahapan ini menjadi titik awal proses karena menentukan arah serta tujuan pekerjaan yang akan dilakukan.

Setelah permintaan diterima, langkah berikutnya adalah melakukan analisis untuk memahami kebutuhan secara menyeluruh. Pada tahap ini ditentukan jenis data yang diperlukan, sistem yang perlu diakses, struktur tabel yang relevan, serta bentuk analisis atau perbandingan data yang diminta. Pemahaman yang tepat pada tahap awal ini menjadi hal penting agar proses teknis yang dilakukan selanjutnya benar-benar selaras dengan tujuan dan tidak menghasilkan data yang keliru.

Tahap berikutnya adalah penyusunan serta pelaksanaan *query*. Pada proses ini, logika *query* dirancang agar mampu menghasilkan data yang dibutuhkan, baik untuk keperluan komparasi maupun pengambilan data tertentu sesuai permintaan. Apabila tugas yang diberikan berupa perbandingan data antar sistem, *query*

disusun untuk mengidentifikasi perbedaan secara rinci, seperti perbedaan nomor identitas atau informasi kontak. Sementara itu, untuk permintaan ekstraksi data, *query* difokuskan pada penerapan filter sesuai dengan parameter yang telah ditentukan. Setelah *query* selesai disusun, proses dilanjutkan dengan eksekusi untuk memperoleh hasil data yang aktual.

Setelah hasil *query* diperoleh, dilakukan tahap validasi serta pengecekan ulang. Pada tahap ini diperiksa apakah hasil perbandingan telah mencerminkan kondisi data yang sebenarnya, apakah relasi antar tabel sudah sesuai, serta apakah data yang dihasilkan memenuhi kriteria permintaan. Untuk hasil ekstraksi data, kelengkapan informasi juga diperiksa agar data siap digunakan oleh pihak yang membutuhkan. Proses validasi ini dilakukan untuk memastikan bahwa data yang disampaikan memiliki tingkat akurasi yang dapat dipertanggungjawabkan.

Apabila data telah melalui proses pengecekan dan dinyatakan valid, hasil tersebut disampaikan kepada atasan. Pada jenis task by request, hasil pekerjaan umumnya tidak memerlukan penyusunan laporan atau ringkasan tambahan karena sifat permintaannya yang cenderung cepat dan spesifik. Data biasanya dikirim dalam bentuk file Excel atau CSV sehingga dapat langsung digunakan oleh atasan atau diteruskan kepada unit kerja terkait lainnya.

Untuk beberapa jenis task tertentu yang membutuhkan penambahan data ke dalam database, proses dilanjutkan ke tahap persiapan data sebelum masuk ke alur ETL. Data hasil *query* disusun kembali agar sesuai dengan struktur tabel tujuan. Selanjutnya, data diproses melalui alur ETL berbasis Python, di mana sistem akan membaca file, melakukan validasi internal, serta memasukkan data ke dalam database. Tahapan ini dilakukan untuk memastikan data tambahan masuk dengan format yang konsisten dan sesuai dengan standar sistem.

Secara keseluruhan, alur kerja task by request menggambarkan proses yang dimulai dari penerimaan permintaan, analisis kebutuhan, penyusunan dan eksekusi *query*, validasi hasil, penyampaian data kepada atasan, hingga proses penambahan data melalui ETL untuk jenis task tertentu. Alur ini menunjukkan bahwa setiap permintaan diproses secara sistematis dan terstruktur, serta selaras

dengan standar kerja yang diterapkan oleh tim *Data Governance*.

3.3 Uraian Pelaksanaan Kerja

Pada pelaksanaan magang, peserta magang bertugas untuk menjalankan tugas *daily task* dan juga memenuhi kebutuhan berdasarkan permintaan tim, yang diuraikan sebagai berikut.

3.3.1 Tahap *Onboarding*

Pada tahap awal pelaksanaan magang, dijalani proses adaptasi melalui kegiatan *Onboarding* di Divisi *Data Governance*. Kegiatan ini mencakup pengenalan struktur organisasi perusahaan serta pemahaman alur kerja yang diterapkan dalam tim. Pada tahap ini, Terjadi pembagian peran antara tim yang menangani sistem SAP CX dan SAP S/4HANA untuk memahami tanggung jawab masing-masing tim serta hubungan kerja dalam pengelolaan data perusahaan.

Selain itu, dipelajari mengenai Standar Operasional Prosedur (SOP) validasi data sebagai acuan dalam menjaga kualitas dan konsistensi data. Dilakukan juga peninjauan awal terhadap struktur tabel data customer guna memahami mekanisme penyimpanan dan pengelolaan data di dalam sistem. Pemahaman tersebut menjadi dasar penting dalam mendukung proses validasi dan analisis data pada tahapan pekerjaan selanjutnya.

3.3.2 Tahap Pembelajaran

Pada tahap pembelajaran, dimulai untuk mendalami berbagai jenis kesalahan data yang umum ditemukan pada sistem SAP CX. Jenis kesalahan tersebut meliputi ketidakkonsistenan kapitalisasi penulisan nama, penggunaan alamat email yang tidak sesuai format, perbedaan penulisan nomor telepon, serta ketidaktepatan pengisian data NPWP. Pemahaman terhadap kategori kesalahan ini menjadi langkah awal yang penting untuk mengenali pola error yang berpotensi memengaruhi kualitas dan keandalan data pelanggan.

Selain mempelajari kategori kesalahan data, dimulai juga penyusunan sekaligus menjalankan *query* sederhana yang digunakan dalam proses

validasi. Pada tahap ini, dilakukan latihan untuk mendeteksi data duplikat dalam sistem serta terlibat dalam proses pengecekan data KTP. Kegiatan tersebut dilakukan untuk memastikan kelengkapan dan kesesuaian data identitas pelanggan, sekaligus mendukung upaya menjaga kualitas data secara berkelanjutan.

3.3.3 Uraian Kerja Daily Task

Dalam pelaksanaan daily task, digunakan *query* yang telah tersedia dan sebelumnya telah melalui proses join dengan beberapa tabel terkait. *Query* tersebut kemudian dikembangkan untuk menghasilkan data profiling, sebagaimana ditunjukkan pada Gambar 3.3.

```
-- Validasi Nama
CASE
    -- 1. Nama individual mengandung kata organisasi
    WHEN f.Customer_Type = 'Individual'
        AND (
            LOWER(f.Account) LIKE '%pt %'
            OR LOWER(f.Account) LIKE '%cv %'
            OR LOWER(f.Account) LIKE '%pt. %'
            OR LOWER(f.Account) LIKE '%pt.%'
            OR LOWER(f.Account) LIKE '%masjid %'
            OR LOWER(f.Account) LIKE '%sekolah %'
        )
        AND LOWER(f.Account) NOT IN ('cv arthur taurosa')
        THEN 'NAMA INDIVIDUAL MENGANDUNG PT. '

    -- 2. Nama null
    WHEN COALESCE(f.Account, '') = ''
        THEN 'NAMA NULL. '

    -- 3. Nama terlalu pendek atau terlalu panjang
    WHEN LEN(f.Account) < 3 OR LEN(f.Account) > 40
        THEN 'NAMA KURANG DARI 3 KARAKTER ATAU LEBIH DARI 40 KARAKTER. '

    -- 4. Nama mengandung kata tidak valid
    WHEN LOWER(f.Account) LIKE 'ibu %'
        OR LOWER(f.Account) LIKE 'bapak %'
        OR LOWER(f.Account) LIKE 'bu %'
        OR LOWER(f.Account) LIKE 'pak %'
        OR LOWER(f.Account) LIKE 'ayah %'
        OR LOWER(f.Account) LIKE 'mama %'
        OR LOWER(f.Account) LIKE '%prank call%'
        OR LOWER(f.Account) LIKE '%mute call %'
        OR LOWER(f.Account) LIKE '%no name %'
        OR LOWER(f.Account) LIKE '%not used %'
        OR LOWER(f.Account) LIKE '%facebook %'
        OR LOWER(f.Account) LIKE '%tiktok %'
        OR LOWER(f.Account) LIKE '%instagram %'
        THEN 'NAMA MENGANDUNG KATA TIDAK VALID. '

    -- 5. Nama ada spasi di depan atau belakang
    WHEN f.Account LIKE ' %' OR f.Account LIKE '% '
        THEN 'NAMA MENGANDUNG SPASI DI DEPAN ATAU BELAKANG. '
```

Gambar 3.3.a Query Validasi Nama

```
    -- 6. Nama tidak kapital (dengan pengecualian gelar)
    WHEN f.Account <> UPPER(f.Account)
        AND LOWER(f.Account) NOT LIKE 'dr. %'
        AND LOWER(f.Account) NOT LIKE 'drs. %'
        AND LOWER(f.Account) NOT LIKE 'dra. %'
        AND LOWER(f.Account) NOT LIKE 'drg. %'
        AND LOWER(f.Account) NOT LIKE 'ir. %'
        AND LOWER(f.Account) NOT LIKE 'sp. %'
        AND LOWER(f.Account) NOT LIKE 'h. %'
        AND LOWER(f.Account) NOT LIKE 'hj. %'
        AND LOWER(f.Account) NOT LIKE 'kh. %'
        AND LOWER(f.Account) NOT LIKE 'ust. %'
        THEN 'NAMA TIDAK KAPITAL. '

    -- 7. Organization dengan tilde di awal
    WHEN f.Customer_Type = 'Organization' AND f.Account LIKE '~%'
        THEN 'NAMA DIAWALI DENGAN TILDE (~). '

    -- 8. Nama mengandung numeric atau special character
    WHEN PATINDEX('%[^a-zA-Z0-9'' .,]%', f.Account) > 0
        THEN 'NAMA MENGANDUNG NUMERIC ATAU SPECIAL CHARACTER. '

    -- 9. Nama mengandung lebih dari 1 nama
    WHEN f.Account LIKE '%QQ%'
        OR f.Account LIKE '% ATAU %'
        OR f.Account LIKE '% OR %'
        OR f.Account LIKE '%&%'
        OR f.Account LIKE '% dan %'
        OR f.Account LIKE '% DAN %'
        OR f.Account LIKE '% and %'
        OR f.Account LIKE '% AND %'
        THEN 'NAMA MENGANDUNG LEBIH DARI 1 NAMA'

    ELSE NULL
END AS Validation_Name
```

Gambar 3.3.b Query Validasi Nama

Bagian validasi nama yang ditampilkan pada Gambar 3.3.a dan 3.3.b berfungsi untuk memastikan bahwa Account Name ditulis sesuai dengan standar penamaan yang berlaku. Tahapan awal validasi dilakukan dengan mengidentifikasi jenis pelanggan. Apabila pelanggan bertipe individual, namun nama yang tercatat mengandung istilah yang lazim digunakan untuk organisasi, seperti “PT”, “CV”, “Masjid”, atau “Sekolah” (dengan beberapa pengecualian tertentu), maka kondisi tersebut dianggap sebagai indikasi kesalahan penginputan karena format nama tidak sesuai dengan jenis pelanggan.

Proses validasi selanjutnya mencakup pemeriksaan terhadap kelengkapan dan kewajaran panjang nama. Sistem akan menandai data apabila kolom nama tidak terisi (null) atau memiliki jumlah karakter di luar batas yang ditetapkan, yaitu kurang dari tiga karakter atau melebihi empat puluh karakter. Selain itu, dilakukan pula pengecekan terhadap keberadaan kata atau frasa yang tidak layak digunakan sebagai nama, seperti sapaan (“Ibu”, “Bapak”, “Bu”, “Pak”), istilah hubungan keluarga (“Ayah”, “Mama”), maupun penanda data tidak valid atau bersifat sementara, seperti “prank call”, “mute call”, “no name”, “not used”, serta nama platform media sosial.

Tahap berikutnya mencakup pemeriksaan teknis terhadap format

penulisan nama, termasuk keberadaan spasi di awal atau akhir teks serta kesesuaian penggunaan huruf kapital. Penulisan nama diharapkan mengikuti format kapitalisasi yang sesuai dengan standar, dengan pengecualian untuk gelar tertentu seperti “Dr.”, “Ir.”, “H.”, dan “Hj.” yang memiliki ketentuan khusus. Untuk akun bertipe organisasi, sistem juga melakukan pengecekan terhadap penggunaan karakter awal yang tidak semestinya, seperti tanda tilde (~).

Pada tahap akhir, sistem mendeteksi apakah nama mengandung angka atau simbol yang tidak diperbolehkan, serta memastikan bahwa satu kolom nama tidak memuat lebih dari satu identitas. Indikasi tersebut dapat berupa penggunaan kata “atau”, “or”, simbol “&”, maupun pola tertentu seperti “QQ”. Seluruh rangkaian aturan validasi ini digunakan untuk mengidentifikasi data nama yang tidak sesuai dengan standar, sehingga dapat ditindaklanjuti dalam proses pembersihan dan perbaikan data.

```
,
-- Validasi KTP
CASE
    WHEN f.Customer_Type != 'Individual'
        THEN CASE WHEN ISNULL(f.No_KTP, '') != '' THEN 'CORPORATE MEMILIKI KTP' ELSE NULL END
    ELSE
        CASE
            WHEN COALESCE(f.No_KTP, '') = '' OR COALESCE(f.No_KTP, '') = 'None'
                THEN 'KTP NULL. '
            WHEN LEN(f.No_KTP) <> 16
                THEN 'KTP TIDAK TERDIRI DARI 16 DIGIT. '
            WHEN f.No_KTP LIKE '%[^0-9]%'
                THEN 'KTP HARUS NUMERIC. '
            WHEN RIGHT(f.No_KTP, 3) = '000'
                THEN '3 DIGIT TERAKHIR KTP ADALAH 000. '
            WHEN LEFT(f.No_KTP, 2) = '00'
                THEN 'NOMOR ID KTP DIMULAI DARI 00. '
            WHEN LEFT(f.No_KTP, 4) = '1234'
                THEN 'NOMOR ID KTP 4 DIGIT PERTAMA MENGANDUNG 1234. '
            ELSE NULL
        END
END AS Validation_NoKTP
,
```

Gambar 3.4 Query Validasi KTP

Validasi nomor KTP sebagaimana ditunjukkan pada Gambar 3.4 dilakukan untuk memastikan bahwa data identitas yang tersimpan di dalam sistem telah sesuai dengan ketentuan penulisan serta karakteristik Nomor Induk Kependudukan yang berlaku. Mekanisme validasi ini dibedakan berdasarkan jenis pelanggan, yaitu pelanggan individual dan pelanggan organization, dengan tujuan menjaga kesesuaian tipe data serta mengurangi potensi kesalahan penginputan pada database master.

Pada pelanggan bertipe organization, proses validasi difokuskan pada

pemeriksaan keberadaan nomor KTP yang seharusnya tidak diisi. Apabila sistem mendeteksi adanya nilai nomor KTP pada entitas corporate, maka data tersebut ditandai sebagai tidak sesuai karena tidak selaras dengan karakteristik tipe pelanggan yang bersangkutan.

Sementara itu, pada pelanggan bertipe individual, validasi dilakukan melalui beberapa tahap pemeriksaan. Sistem terlebih dahulu mengecek apakah kolom nomor KTP kosong atau berisi nilai yang tidak semestinya, seperti "None". Selanjutnya dilakukan pengecekan panjang nomor KTP yang harus terdiri dari enam belas digit. Nomor dengan jumlah digit kurang atau melebihi ketentuan tersebut dikategorikan sebagai tidak valid. Selain itu, seluruh karakter pada nomor KTP harus berupa angka, sehingga keberadaan huruf maupun simbol lain dianggap sebagai kesalahan penginputan.

Validasi juga mencakup pemeriksaan terhadap pola angka tertentu yang dapat mengindikasikan data tidak valid secara struktural, seperti tiga digit terakhir bernilai 000, dua digit awal bernilai 00, atau empat digit pertama yang diawali dengan angka 1234. Pola-pola tersebut digunakan sebagai indikator tambahan untuk mendeteksi nomor KTP yang tidak sesuai dengan standar. Apabila seluruh kriteria validasi terpenuhi dan tidak ditemukan pelanggaran aturan, maka data nomor KTP dinyatakan valid dan tidak diberikan penanda kesalahan.

```
-- Validasi NPWP16
CASE
    WHEN f.Customer_Type IN ('Organization','Individual')
        THEN CASE
            WHEN COALESCE(f.NPWP16, '') = '' OR f.NPWP16 = 'None'
                THEN 'NPWP16 NULL. '
            WHEN f.NPWP16 LIKE '%0000000000000000%'
                THEN 'NPWP CONTAIN ALL ZERO. '
            WHEN LEN(f.NPWP16) <> 16
                THEN 'NPWP16 TIDAK TERDIRI DARI 16 DIGIT. '
            WHEN f.NPWP16 LIKE '%[^0-9]%'
                THEN 'NPWP16 HARUS NUMERIC. '
            ELSE NULL
        END
    ELSE NULL
END AS Validation_NoNPWP16
,
-- Validasi NPWP15
CASE
    WHEN f.Customer_Type IN ('Organization','Individual')
        THEN CASE
            WHEN LOWER(LTRIM(RTRIM(COALESCE(f.No_NPWP, '')))) IN ('','none')
                THEN 'NPWP15 NULL. '
            WHEN LEN(REPLACE(REPLACE(REPLACE(f.No_NPWP,'.',''),'-',''),' ','')) <> 15
                THEN 'NPWP15 TIDAK TERDIRI DARI 15 DIGIT. '
            WHEN f.No_NPWP NOT LIKE '[0-9][0-9].[0-9][0-9][0-9].[0-9][0-9][0-9].[0-9]-[0-9][0-9][0-9].[0-9][0-9][0-9]'
                AND PATINDEX('[0-9]{15}', f.No_NPWP) <> 1
                AND COALESCE(f.No_NPWP,'') <> ''
                THEN 'NPWP15 FORMAT TIDAK VALID. '
            ELSE NULL
        END
    ELSE NULL
END AS Validation_NoNPWP15
```

Gambar 3.5 Query Validasi NPWP

Query yang ditunjukkan pada Gambar 3.5 digunakan untuk melakukan pengecekan kelengkapan serta keabsahan nomor NPWP pada customer bertipe Organization maupun Individual. Pada validasi NPWP16, sistem menerapkan sejumlah aturan secara berurutan, dimulai dengan memastikan bahwa kolom NPWP16 tidak bernilai kosong dan tidak berisi nilai "None". Selanjutnya, dilakukan pemeriksaan untuk memastikan bahwa nomor NPWP tidak berupa rangkaian angka nol seluruhnya. Sistem juga memverifikasi bahwa panjang NPWP16 terdiri dari enam belas digit serta memastikan seluruh karakter yang digunakan bersifat numerik tanpa mengandung huruf maupun simbol lainnya. Apabila salah satu ketentuan tersebut tidak terpenuhi, maka *query* akan menghasilkan remark yang menjelaskan jenis kesalahan yang ditemukan.

Untuk NPWP15, proses validasi dilakukan melalui tahapan yang serupa, namun disesuaikan dengan karakteristik format NPWP lima belas digit. Tahap awal memastikan bahwa nilai NPWP tidak kosong dan tidak berisi "None". Selanjutnya, karakter tanda baca seperti titik, tanda hubung, dan spasi dihilangkan terlebih dahulu untuk memperoleh nilai numerik murni. Setelah proses pembersihan tersebut, sistem memeriksa apakah panjang data yang dihasilkan berjumlah lima belas digit serta mengecek kesesuaian format dengan pola resmi NPWP, yaitu xx.xxx.xxx.x-xxx.xxx. Apabila tanda baca tidak digunakan, nilai NPWP tetap harus berupa lima belas digit angka yang berurutan. Jika seluruh ketentuan format tersebut tidak terpenuhi, sistem akan memberikan remark "NPWP15 FORMAT TIDAK VALID" sebagai penanda bahwa penulisan NPWP tidak sesuai dengan aturan yang berlaku.

```
,
-- Validasi Email
CASE
    WHEN f.Customer_Type IN ('Organization','Individual')
        THEN CASE
            WHEN COALESCE(f.EMail, '') = '' OR f.EMail = 'None'
                THEN 'EMAIL NULL. '
            WHEN f.EMail LIKE 'xxx%' OR f.EMail LIKE 'XXX%'
                THEN 'EMAIL DIMULAI DARI XXX. '
            WHEN PATINDEX('%[^a-zA-Z0-9@._-]%', f.EMail) > 0
                THEN 'EMAIL MENGANDUNG KARAKTER TIDAK VALID. '
            WHEN LEN(f.EMail) < 5
                THEN 'EMAIL KURANG DARI 5 KARAKTER. '
            WHEN f.EMail NOT LIKE '%@%.%' OR f.EMail LIKE '%000@000.000'
                THEN 'EMAIL TIDAK VALID. '
            ELSE NULL
        END
    ELSE NULL
END AS Validation_Email
```

Gambar 3.6 Query Validasi Email

Validasi email sebagaimana ditunjukkan pada Gambar 3.6 dilakukan untuk memastikan bahwa alamat email yang tersimpan pada data pelanggan telah memenuhi standar penulisan yang berlaku serta terbebas dari kesalahan umum. Proses ini diterapkan pada pelanggan bertipe Individual maupun Organization. Tahap awal pemeriksaan dimulai dengan mengecek apakah kolom email kosong atau berisi nilai “None”, karena kondisi tersebut menunjukkan bahwa data email belum tersedia. Selain itu, sistem juga mengidentifikasi alamat email yang diawali dengan pola seperti “xxx” atau “XXX” yang umumnya digunakan sebagai placeholder dan tidak layak disimpan pada database operasional.

Tahap berikutnya berfokus pada pemeriksaan karakter yang digunakan dalam alamat email. Sistem memastikan bahwa email hanya terdiri dari karakter yang diperbolehkan, yaitu huruf, angka, simbol at (@), titik, garis bawah, dan tanda hubung. Panjang email juga diperiksa untuk memastikan telah memenuhi batas minimal karakter agar dapat dianggap valid secara format. Setelah itu, dilakukan pengecekan terhadap struktur dasar email dengan memastikan keberadaan simbol at dan titik pada posisi yang sesuai. Alamat email dengan pola yang tidak memenuhi struktur tersebut, termasuk format tidak wajar seperti “000@000.000”, akan dikategorikan sebagai email tidak valid.

Melalui rangkaian pemeriksaan tersebut, berbagai potensi kesalahan pada data email dapat teridentifikasi, mulai dari ketiadaan data, penggunaan placeholder, karakter yang tidak sesuai, panjang email yang tidak memadai,

hingga kesalahan struktur penulisan. Validasi ini membantu menjaga kualitas data email agar tetap dapat digunakan secara optimal dalam mendukung kebutuhan operasional yang memerlukan keakuratan informasi kontak pelanggan.

```
,
-- Validasi Mobile
CASE
    WHEN f.Customer_Type IN ('Organization','Individual')
        THEN CASE
            WHEN COALESCE(f.Mobile,'') = '' OR f.Mobile = 'None'
                THEN 'HP NULL. '
            WHEN f.Mobile NOT LIKE '62%'
                AND f.Mobile NOT LIKE '02%'
                AND f.Mobile NOT LIKE '03%'
                AND f.Mobile NOT LIKE '08%'
                AND f.Mobile NOT LIKE '+%'
                THEN 'HP HARUS DIAWALI DENGAN +62 ATAU 08 ATAU 02 ATAU 03. '
            WHEN f.Mobile LIKE '%0000%'
                THEN 'HP MENGANDUNG 0000. '
            WHEN LEN(f.Mobile) < 8 OR LEN(f.Mobile) > 15
                THEN 'HP HARUS TERDIRI DARI 8-15 DIGIT. '
            WHEN f.Mobile LIKE '%+62 62%' OR f.Mobile LIKE '%+6262%'
                THEN 'HP MENGANDUNG +62 62. '
            WHEN PATINDEX('%[A-Za-z]%', f.Mobile) > 0
                THEN 'HP TIDAK BOLEH MENGANDUNG HURUF. '
            ELSE NULL
        END
    ELSE NULL
END AS Validation_Mobile
```

Gambar 3.7 Query Validasi Mobile

Melalui Validasi mobile sebagaimana ditunjukkan pada Gambar 3.7 dilakukan untuk memastikan bahwa nomor telepon yang tersimpan pada database ditulis dengan format yang benar, tidak menggunakan nilai placeholder, serta sesuai dengan standar penomoran yang berlaku. Proses validasi ini diterapkan pada pelanggan bertipe Individual maupun Organization.

Tahap awal pemeriksaan dimulai dengan mengecek apakah nilai mobile kosong atau berisi “None”, karena kondisi tersebut menunjukkan bahwa data nomor telepon belum tersedia. Selanjutnya, sistem melakukan verifikasi terhadap awalan nomor telepon. Nomor yang dianggap valid harus diawali dengan kode internasional atau kode area tertentu, seperti 62, 02, 03, 08, atau tanda plus (+). Apabila awalan nomor tidak sesuai dengan ketentuan tersebut, maka nomor dikategorikan sebagai tidak valid.

Pemeriksaan berikutnya difokuskan pada identifikasi pola angka yang tidak wajar. Nomor yang mengandung urutan angka “0000” akan ditandai sebagai tidak valid karena pola tersebut tidak lazim digunakan dalam penomoran pelanggan. Selain itu, panjang nomor juga divalidasi dan harus

berada dalam rentang delapan hingga lima belas digit. Nomor yang berada di luar rentang tersebut dianggap tidak memenuhi standar. Validasi tambahan dilakukan untuk mendeteksi kesalahan pengulangan awalan, seperti pola +62 62 atau +6262, yang mengindikasikan adanya kesalahan input kode negara.

Tahap terakhir memastikan bahwa nilai mobile hanya terdiri dari karakter numerik. Apabila ditemukan karakter huruf pada nomor telepon, maka data tersebut dinyatakan tidak valid. Melalui rangkaian pemeriksaan ini, validasi mobile membantu menjaga kualitas data nomor telepon agar dapat digunakan secara konsisten dalam proses operasional maupun integrasi sistem yang memerlukan keakuratan informasi kontak pelanggan.

```
    ,
    -- Validasi Birthdate
  CASE
 WHEN f.Customer_Type = 'Individual' THEN
    CASE
        WHEN f.Date_of_Birth IS NULL
             OR LTRIM(RTRIM(CONVERT(VARCHAR(50), f.Date_of_Birth))) = ''
             OR LTRIM(RTRIM(CONVERT(VARCHAR(50), f.Date_of_Birth))) IN ('NULL','Null','null')
             OR LEN(LTRIM(RTRIM(CONVERT(VARCHAR(50), f.Date_of_Birth)))) = 0
            THEN 'BIRTHDATE NULL. '
        WHEN ISDATE(CONVERT(VARCHAR(50), f.Date_of_Birth)) = 1
             AND YEAR(CAST(f.Date_of_Birth AS DATE)) = 9999
            THEN 'TANGGAL LAHIR MENGANDUNG TAHUN 9999. '
        WHEN ISDATE(CONVERT(VARCHAR(50), f.Date_of_Birth)) = 1
             AND CAST(f.Date_of_Birth AS DATE) IN (CAST('1700-01-01' AS DATE), CAST('1900-01-01' AS DATE))
            THEN 'TANGGAL LAHIR MENGANDUNG TAHUN TIDAK VALID. '
        ELSE NULL
    END

 WHEN f.Customer_Type = 'Organization' THEN
    CASE
        WHEN f.Date_of_Birth IS NULL
             OR LTRIM(RTRIM(CONVERT(VARCHAR(50), f.Date_of_Birth))) = ''
             OR LTRIM(RTRIM(CONVERT(VARCHAR(50), f.Date_of_Birth))) IN ('NULL','Null','null')
             OR LEN(LTRIM(RTRIM(CONVERT(VARCHAR(50), f.Date_of_Birth)))) = 0
            THEN 'ESTABLISHMENT DATE NULL. '
        WHEN ISDATE(CONVERT(VARCHAR(50), f.Date_of_Birth)) = 1
             AND YEAR(CAST(f.Date_of_Birth AS DATE)) = 9999
            THEN 'ESTABLISHMENT DATE MENGANDUNG TAHUN 9999. '
        WHEN ISDATE(CONVERT(VARCHAR(50), f.Date_of_Birth)) = 1
             AND CAST(f.Date_of_Birth AS DATE) IN (CAST('1700-01-01' AS DATE), CAST('1900-01-01' AS DATE))
            THEN 'ESTABLISHMENT DATE MENGANDUNG TAHUN TIDAK VALID. '
        ELSE NULL
    END
    |
  ELSE NULL
END AS Validation_Birthdate
```

Gambar 3.8 Query Validation BirthDate

Validasi birthdate pada Gambar 3.8 bertujuan untuk memastikan bahwa data tanggal lahir pada pelanggan individual atau tanggal pendirian pada pelanggan organization terisi dengan lengkap dan masuk akal secara logis. Tahap awal validasi dilakukan dengan memeriksa apakah nilai tanggal kosong, tidak terisi, atau hanya berisi representasi teks seperti NULL, Null, atau none. Kondisi tersebut menunjukkan bahwa informasi tanggal tidak tersedia, sehingga sistem akan memberikan penanda khusus.

Selanjutnya, dilakukan identifikasi terhadap tanggal yang secara

format terlihat valid, namun tidak wajar dari sisi bisnis. Beberapa nilai yang sering muncul pada sistem SAP, seperti 1700-01-01 dan 1900-01-01, diketahui sebagai default system date atau placeholder yang digunakan ketika data tanggal tidak pernah diinput, terjadi kegagalan migrasi data, atau field tanggal tidak memiliki nilai saat proses pemrosesan berjalan. Karena tanggal-tanggal tersebut tidak merepresentasikan tanggal lahir maupun tanggal pendirian yang realistis, maka nilai tersebut diklasifikasikan sebagai tidak valid.

Selain itu, validasi juga mencakup pemeriksaan terhadap nilai tahun 9999, yang pada beberapa sistem backend digunakan sebagai penanda maksimum tanggal. Nilai ini menunjukkan bahwa data tanggal belum ditentukan dan tidak dapat dianggap sebagai informasi yang valid. Melalui rangkaian pemeriksaan tersebut, proses validasi memastikan bahwa data tanggal yang tersimpan bersifat realistis, relevan, dan dapat digunakan secara andal dalam proses analisis maupun pengolahan data lanjutan.

```
,
-- Validasi Gender
CASE
    WHEN f.Customer_Type = 'Individual'
        THEN CASE
            WHEN COALESCE(f.Gender,'') = '' OR f.Gender = '0'
                THEN 'GENDER NULL. '
            WHEN f.Gender LIKE '% %'
                THEN 'GENDER MENGANDUNG SPASI. '
            WHEN f.Gender NOT IN ('1','2')
                THEN 'GENDER BUKAN 1 ATAU 2. '
            ELSE NULL
        END
    ELSE NULL
END AS Validation_Gender
```

Gambar 3.9 Query Validasi Gender

Validasi gender pada Gambar 3.9 diterapkan khusus untuk pelanggan dengan kategori Individual. Tahap awal pemeriksaan dilakukan untuk memastikan bahwa field gender terisi dengan benar. Data akan ditandai tidak valid apabila nilainya kosong, berupa string kosong, NULL, maupun angka 0 yang dalam sistem diperlakukan sebagai nilai tidak terisi.

Selanjutnya dilakukan pengecekan format penulisan untuk memastikan bahwa nilai gender tidak mengandung spasi, baik di awal, di tengah, maupun di akhir. Keberadaan spasi menunjukkan bahwa data tidak

disimpan sesuai dengan format standar yang ditetapkan.

Tahap terakhir adalah verifikasi nilai gender berdasarkan ketentuan internal sistem. Nilai yang diperbolehkan hanya 1 dan 2, di mana angka 1 merepresentasikan laki-laki dan angka 2 merepresentasikan perempuan. Nilai di luar dua kode tersebut dikategorikan sebagai tidak valid karena tidak sesuai dengan standar yang berlaku. Melalui rangkaian pemeriksaan ini, data gender dapat tercatat secara konsisten dan siap digunakan dalam proses analisis maupun integrasi antar sistem.

```
,
-- Validasi Address
CASE
    WHEN f.street_cust IS NULL OR f.street_cust = ''
        THEN 'ADDRESS NULL. '
    WHEN LEN(f.street_cust) < 4
        THEN 'ADDRESS <4 CHAR. '
    ELSE NULL
END AS Validation_Address
```

Gambar 3.10 Query Validasi Address

Validasi alamat seperti pada Gambar 3.10 dilakukan untuk memastikan bahwa informasi alamat pada kolom street_cust telah terisi dan memiliki kualitas minimum yang dapat digunakan. Tahap awal pemeriksaan difokuskan pada ketersediaan data alamat. Apabila nilai street_cust bernilai NULL atau hanya berisi teks kosong, maka data tersebut dinyatakan tidak memiliki alamat dan ditandai sebagai "ADDRESS NULL". Kondisi ini dianggap bermasalah karena ketiadaan alamat dapat menghambat proses identifikasi pelanggan maupun aktivitas operasional lainnya.

Tahap selanjutnya adalah pemeriksaan panjang alamat. Alamat dengan jumlah karakter kurang dari empat dianggap tidak representatif dan tidak memberikan informasi yang memadai. Data dengan kondisi tersebut ditandai sebagai "ADDRESS <4 CHAR", karena alamat yang terlalu singkat umumnya tidak dapat digunakan secara efektif untuk keperluan validasi lanjutan, pengiriman, ataupun pemetaan lokasi.

Validation_Name	Validation_NoKTP	Validation_NoNPWP16	Validation_NoNPWP15	Validation_Email	Validation_Mobile	Validation_Birthdate	Validation_Gender	Validation_Address
NULL	NULL	NULL	NPWP15 NULL.	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NPWP16 NULL.	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NPWP16 NULL.	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	HP NULL.	NULL	NULL	NULL
NULL	NULL	NPWP16 NULL.	NPWP15 NULL.	NULL	NULL	ESTABLISHMENT DATE NULL.	NULL	ADDRESS NULL.
NULL	NULL	NPWP16 NULL.	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NPWP16 NULL.	NPWP15 NULL.	NULL	NULL	ESTABLISHMENT DATE NULL.	NULL	ADDRESS NULL.
NULL	NULL	NULL	NULL	NULL	HP NULL.	NULL	NULL	NULL
NULL	NULL	NULL	NPWP15 NULL.	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	EMAIL TIDAK VALID.	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NPWP15 NULL.	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NPWP15 NULL.	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NPWP15 NULL.	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NPWP16 NULL.	NPWP15 NULL.	NULL	NULL	ESTABLISHMENT DATE NULL.	NULL	ADDRESS NULL.
NULL	NULL	NULL	NPWP15 NULL.	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NPWP15 NULL.	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NPWP16 NULL.	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NPWP16 NULL.	NPWP15 NULL.	NULL	NULL	ESTABLISHMENT DATE NULL.	NULL	ADDRESS NULL.
NULL	NULL	NPWP16 NULL.	NULL	NULL	NULL	NULL	NULL	NULL

Gambar 3.11 Hasil Query Daily Task

Pada Gambar 3.11 ditampilkan hasil *query* Daily Task yang menyajikan status validasi untuk setiap atribut data pelanggan, meliputi nama, KTP, NPWP16, NPWP15, email, nomor telepon, tanggal lahir atau establishment date, gender, serta alamat. Cara membaca hasil ini cukup sederhana, yaitu nilai "NULL" pada kolom validasi menunjukkan bahwa atribut tersebut telah memenuhi ketentuan, sedangkan kemunculan teks tertentu menandakan adanya ketidaksesuaian terhadap aturan validasi yang diterapkan pada *query*. Dalam konteks ini, nilai "NULL" tidak merujuk pada kondisi data kosong di database, melainkan menunjukkan bahwa proses evaluasi tidak menemukan kesalahan pada atribut yang bersangkutan. Sebaliknya, pesan seperti "NPWP15 NULL", "EMAIL TIDAK VALID", "HP NULL", atau "ESTABLISHMENT DATE NULL" merupakan hasil dari logika validasi yang secara spesifik menandai jenis permasalahan yang ditemukan pada masing-masing kolom data.

Perbedaan hasil validasi juga dapat diamati antara pelanggan bertipe Individual dan Organization, khususnya pada kolom tanggal. Untuk pelanggan Individual, sistem melakukan pemeriksaan terhadap birthdate dan akan menghasilkan pesan kesalahan apabila tanggal lahir tidak tersedia, menggunakan nilai default sistem yang tidak wajar seperti 1700-01-01 atau 1900-01-01, maupun mengandung tahun 9999 yang umumnya berasal dari error atau nilai teknis SAP. Sementara itu, untuk pelanggan bertipe Organization, kolom yang sama diperlakukan sebagai establishment date, sehingga pesan validasi yang muncul disesuaikan, misalnya "ESTABLISHMENT DATE NULL" atau "ESTABLISHMENT DATE

MENGANDUNG TAHUN 9999". Perbedaan ini menunjukkan bahwa *query* telah dirancang secara kontekstual dengan mempertimbangkan karakteristik data sesuai tipe pelanggan.

Secara keseluruhan, tabel hasil *query* ini berperan sebagai ringkasan kondisi kualitas data harian yang memudahkan tim dalam mengidentifikasi atribut mana yang sudah memenuhi standar dan mana yang masih memerlukan perbaikan. Pola interpretasi "NULL berarti valid, sedangkan adanya keterangan berarti tidak valid" memberikan kejelasan dalam membaca hasil, sementara pesan kesalahan yang spesifik membantu proses tindak lanjut karena secara langsung menunjukkan jenis kesalahan, lokasi atribut, serta tipe pelanggan terkait. Dengan penyajian hasil seperti ini, proses pemeriksaan dan data cleansing dapat dilakukan secara lebih terstruktur, transparan, dan mudah ditelusuri pada tahap selanjutnya.

24/09/2025	Run BAU 1	Validasi data CX valid/invalid (23 Sept 2025)	Pass	Pass	Pass
25/09/2025	Run BAU 1	Validasi data CX valid/invalid (24 Sept 2025)	Pass	Pass	Pass
26/09/2025	Run BAU 1	Validasi data CX valid/invalid (25 Sept 2025)	Pass	Pass	Pass
29/09/20265	Run BAU 1	Validasi data CX valid/invalid (26-28 Sept 2025)	Mobile Null	Pending	[redacted] (26-09-2025)
30/09/2025	Run BAU 1	Validasi data CX valid/invalid (29 Sept 2025)	Mobile Null	Solved	[redacted]
1/10/2025	Run BAU 1	Validasi data CX valid/invalid (30 Sept 2025)	Pass	Pass	Pass
2/10/2025	Run BAU 1	Validasi data CX valid/invalid (01 Oct 2025)	Pass	Pass	Pass
3/10/2025	Run BAU 1	Validasi data CX valid/invalid (02 Oct 2025)	Pass	Pass	Pass
6/10/2025	Run BAU 1	Validasi data CX valid/invalid (03-05 Oct	Pass	Pass	Pass

Gambar 3. 12 Pencatatan dan Laporan

Tabel pada Gambar 3.12 menampilkan pencatatan harian untuk proses BAU 1 dalam validasi data SAP CX. Setiap baris mencatat tanggal pelaksanaan, jenis aktivitas yang dilakukan, hasil validasi, serta status perbaikan apabila ditemukan anomali. Secara umum, mayoritas hasil validasi menunjukkan "Pass", yang menandakan bahwa data pada hari tersebut telah sesuai dengan kriteria. Pada beberapa hari tertentu, terdapat temuan seperti "Mobile Null", yang berarti nomor telepon pelanggan tidak terisi. Temuan ini dicatat beserta status tindak lanjutnya, misalnya masih "Pending" atau telah "Solved", dan dilengkapi keterangan tambahan seperti tanggal perbaikan. Dengan pencatatan harian seperti ini, proses pemantauan kualitas data berjalan lebih terstruktur dan memudahkan tim dalam menelusuri serta

menindaklanjuti setiap isu yang muncul dari waktu ke waktu.

3.3.4 Uraian Kerja Berdasarkan Permintaan

Pada salah satu permintaan pekerjaan, diberikan permintaan untuk melakukan pengecekan kesesuaian nomor KTP pelanggan yang tersimpan pada dua sistem utama, yaitu MDM (*Master Data Management*) dan CX (SAP Customer Experience). Pengecekan ini bertujuan untuk memastikan bahwa kedua sistem memiliki data KTP yang sama (match) sehingga dapat digunakan secara konsisten untuk kebutuhan operasional, analitik, maupun proses verifikasi identitas. Untuk melakukan pengecekan tersebut, digunakan *query* SQL seperti yang ditampilkan pada Gambar 3.13. *Query* ini berfungsi untuk mengambil data pelanggan dari kedua database dan membandingkan nomor KTP mereka secara langsung.

```
SELECT
      MDM.Source_AccountID AS MDM_AccountID
    , CX.CX_AccountID
    , MDM.FullName AS MDM_FullName
    , MDM.IDKTP AS MDM_KTP
    , CX.NormalisedNoKTP AS CX_KTP
    ,
      CASE
          WHEN MDM.IDKTP = CX.NormalisedNoKTP THEN 'MATCH'
          WHEN CX.NormalisedNoKTP IS NULL OR MDM.IDKTP IS NULL THEN 'INCOMPLETE DATA'
          ELSE 'NOT MATCH'
      END AS STATUS_KTP
FROM SD_100_DE_LH_MDM.dbo.MDM_CUSTOMER_INDIVIDUAL_REF MDM
INNER JOIN (
    SELECT
          C_BpIntId AS CX_AccountID
        , C_x1ANxAfef012272ffb5b AS NormalisedNoKTP
    FROM SD_100_DE_LH_SAPCX_CXT.dbo.CXT_codcustomer
) CX
    ON MDM.Source_AccountID = CX.CX_AccountID;
```

Gambar 3. 13 Query Pengecekan KTP

Gambar tersebut menampilkan *query* SQL yang melakukan join antara tabel customer di MDM dan tabel customer di CX berdasarkan *Account ID* pelanggan. Tujuannya adalah untuk menampilkan pasangan data KTP dari kedua sistem sekaligus menentukan apakah keduanya sesuai. Secara garis besar, *query* ini melakukan tiga hal utama:

a. Mengambil Data Identitas Pelanggan

Bagian awal *query* melakukan *select* terhadap beberapa kolom penting:

1) MDM.Source_AccountID → ID pelanggan di sistem MDM
2) CX.CX_AccountID → ID pelanggan di sistem CX
3) MDM.FullName → nama lengkap pelanggan dari MDM
4) MDM.IDKTP → nomor KTP dari MDM
5) CX.NormalisedNoKTP → nomor KTP dari CX yang sudah dinormalisasi

b. Menentukan Status Kesesuaian KTP

Bagian yang paling penting terdapat pada blok CASE**:**

```
CASE
  WHEN MDM.IDKTP = CX.NormalisedNoKTP THEN 'MATCH'
  WHEN CX.NormalisedNoKTP IS NULL OR MDM.IDKTP IS NULL THEN
'INCOMPLETE DATA'
  ELSE 'NOT MATCH'
END AS STATUS_KTP
```

1) MATCH → Jika nomor KTP pada MDM dan CX sama persis
2) INCOMPLETE DATA → Jika salah satu sistem tidak memiliki nomor KTP (NULL)
3) NOT MATCH → Jika kedua KTP ada, tetapi nilainya berbeda

c. Menghubungkan Data MDM dan CX

Query menggunakan INNER JOIN untuk mencocokkan pelanggan berdasarkan *Account ID*:

```
INNER JOIN (
  SELECT
    C_BpIntId AS CX_AccountID,
    c_xlAnXafe012272ffb5b AS NormalisedNoKTP
  FROM SD_100_DE_LH_SAPCX_CXT.dbo.CXT_codcustomer
) CX
ON MDM.Source_AccountID = CX.CX_AccountID;
```

Sub*query* pada bagian CX mengambil:

1) C_BpIntId → akun pelanggan
2) NoKTP yang telah dinormalisasi untuk memastikan formatnya seragam (tanpa tanda baca atau spasi)

Proses join memastikan bahwa perbandingan dilakukan pada data pelanggan yang benar dan sudah saling terhubung antar sistem. Pengecekan kesesuaian nomor KTP antara sistem MDM dan CX yang dilakukan, menunjukkan pentingnya konsistensi data antar platform dalam mendukung proses operasional dan analitik perusahaan. Melalui *query* SQL yang ditunjukkan pada Gambar 3.13, Ditampilkan bahwa telah berhasil mengambil data pelanggan dari kedua sistem, menormalkan nomor KTP dari CX, serta membandingkannya dengan nomor KTP pada MDM menggunakan logika CASE untuk menentukan apakah datanya MATCH, NOT MATCH, atau INCOMPLETE DATA. Proses ini melibatkan pengambilan identitas dasar pelanggan, penentuan status kesesuaian KTP, hingga penghubungan data antar sistem melalui INNER JOIN berdasarkan Account ID. Sehingga, ditekankan bahwa hasil pengecekan ini memberikan gambaran jelas mengenai kualitas data KTP di kedua sistem dan memperlihatkan bagaimana perbedaan input maupun ketidaklengkapan data memengaruhi integritas informasi pelanggan.

	MDM_AccountID	CX_AccountID	MDM_FullName	MDM_KTP	CX_KTP	STATUS_KTP
1						MATCH
2						MATCH
3						MATCH
4						MATCH
5						MATCH
6						MATCH
7						MATCH
8						MATCH
9						MATCH
10						MATCH
11						MATCH
12						MATCH
13						MATCH
14						MATCH
15						MATCH

Gambar 3. 14 Result Pengecekan KTP

Pada Gambar 3.14 terlihat bahwa setiap baris berisi pasangan data pelanggan antara sistem MDM dan CX. Berdasarkan hasil yang muncul pada tabel tersebut, seluruh baris menampilkan status:

STATUS_KTP = “MATCH”

Hal ini berarti bahwa:

1) Nomor KTP pada MDM dan CX identik, tanpa perbedaan angka ataupun format.
2) Kedua sistem telah tersinkronisasi dengan baik dalam hal penyimpanan data identitas pelanggan.
3) Tidak ditemukan data yang incomplete (NULL) ataupun tidak cocok (NOT MATCH) pada sampel data yang dicek.
4) Proses normalisasi nomor KTP pada database CX terbukti efektif karena format KTP sama dengan format di MDM.

Berdasarkan hasil tersebut, dapat disimpulkan beberapa hal:

1. Konsistensi Data Terjaga

   Data KTP yang tersimpan di dua sistem berbeda telah melalui proses sinkronisasi dan cleansing, sehingga tidak ditemukan perbedaan. Ini menunjukkan bahwa proses integrasi antara MDM dan CX berjalan dengan baik.

2. Tidak Ada Temuan “NOT MATCH”

   Absennya status NOT MATCH menunjukkan bahwa:

   a. Tidak ada kesalahan input nomor KTP di CX
   b. Tidak ada perubahan KTP yang tidak tercatat di MDM
   c. Tidak ada kasus duplikasi identitas pada sampel data
   d. Ini merupakan kondisi ideal dalam proses *Data Governance*.

3. Tidak Ada “INCOMPLETE DATA”

   a. Tidak ditemukannya data kosong atau NULL pada MDM_KTP maupun CX_KTP mengindikasikan bahwa:
   b. Semua pelanggan pada data tersebut memiliki KTP yang lengkap
   c. PIC lapangan memasukkan nomor KTP sesuai prosedur
   d. Tidak ada data pelanggan yang dibuat tanpa KTP

Seluruh baris data menunjukkan STATUS_KTP = “MATCH”, yang berarti nomor KTP pelanggan pada sistem MDM dan CX identik tanpa perbedaan nilai maupun format. Temuan ini menunjukkan bahwa kedua sistem telah tersinkronisasi dengan baik dan mekanisme normalisasi nomor KTP pada database CX berfungsi secara efektif. Tidak ditemukannya data dengan status NOT MATCH maupun INCOMPLETE DATA mengindikasikan bahwa tidak ada kesalahan input, tidak ada perubahan KTP yang tidak tercatat, serta tidak ada duplikasi identitas pada sampel data yang dianalisis. Selain itu, ketiadaan nilai NULL pada kolom KTP dari kedua sistem membuktikan bahwa seluruh pelanggan dalam dataset memiliki data identitas lengkap dan diinput sesuai prosedur PIC lapangan.

Setelah melakukan pengecekan awal terhadap kesesuaian nomor KTP pada database MDM dan CX, langkah berikutnya adalah melakukan analisis kualitas data untuk mengidentifikasi nomor KTP yang tidak valid. Validitas KTP sangat penting karena KTP merupakan identitas utama pelanggan dan digunakan dalam berbagai proses bisnis seperti verifikasi, pengajuan produk, serta integrasi antar sistem. Untuk tujuan tersebut, ditambahkan filter tambahan pada dataset sehingga hanya menampilkan pengguna yang memiliki nomor KTP tidak valid. *Query* ditunjukkan pada Gambar 3.15.

```
SELECT *
FROM data_filtered
WHERE Role = 'CRM000'
  AND Customer_Type = 'Individual'
  AND status_rp_id = 'Ada unit'
  AND No_KTP LIKE '%[^0-9]%'      -- mengandung non-digit (tidak valid)
  AND No_KTP NOT LIKE '[A-Za-z]%'; -- exclude jika diawali huruf (kemungkinan paspor)
```

Gambar 3. 15 Query mencari KTP yang tidak valid

Query ini melakukan filtering berdasarkan beberapa kondisi khusus yang bertujuan untuk mendeteksi KTP yang tidak sesuai standar nasional, yaitu 16 digit numerik tanpa karakter tambahan. Berikut adalah logika dari *query* tersebut:

SELECT *

```
FROM data_filtered
WHERE Role = 'CRM000'
  AND Customer_Type = 'Individual'
  AND status_rp_id = 'Ada unit'
  AND NO_KTP LIKE '%[^0-9]%'
  -- Mengandung non-digit (tidak valid)

  AND NO_KTP NOT LIKE '[A-Za-z]%'
  -- Mengecualikan jika diawali huruf (kemungkinan paspor)
;
```

a. Role = 'CRM000': Hanya mengambil customer dengan peran tertentu, dalam kasus ini adalah *CRM000*, yang biasanya merupakan pelanggan aktif di sistem.
b. Customer_Type = 'Individual': Fokus hanya pada pelanggan individu, bukan badan usaha.
c. status_rp_id = 'Ada unit': *Memastikan hanya pelanggan dengan unit aktif yang dianalisis, sehingga data relevan untuk validasi.*
d. NO_KTP LIKE '%[^0-9]%': Bagian ini adalah inti dari deteksi KTP tidak valid.
    1) Kondisi ini mencari NO_KTP yang mengandung karakter apa pun selain angka (0–9).
    2) Artinya, bila ada huruf, tanda baca, spasi, simbol, atau karakter khusus, maka nomor KTP tersebut dinyatakan tidak valid.

    Contoh karakter non-digit yang sering ditemukan:
    1) Huruf seperti X, O, atau 1
    2) Simbol seperti –, ., #, /Spasi di tengah nomor

    Ini merupakan bentuk kesalahan input yang harus diperbaiki.
e. NO_KTP NOT LIKE '[A-Za-z]%': Kondisi ini digunakan untuk mengecualikan data yang memang bukan KTP, tetapi paspor, karena paspor biasanya diawali huruf. Jika nomor diawali huruf, maka data tersebut tidak dimasukkan ke hasil error, agar tidak salah dikategorikan sebagai KTP invalid.

Hasil dari *query* tersebut hanya menampilkan pelanggan yang

nomor KTP-nya mengandung karakter tidak valid. Berdasarkan gambar hasil yang diberikan, ditemukan beberapa kasus seperti:

a. Nomor KTP mengandung huruf "X"
b. Nomor KTP mengandung simbol seperti "-" (dash)
c. Nomor KTP mengandung tanda "#"
d. Beberapa nomor terdiri atas gabungan angka dan non-angka

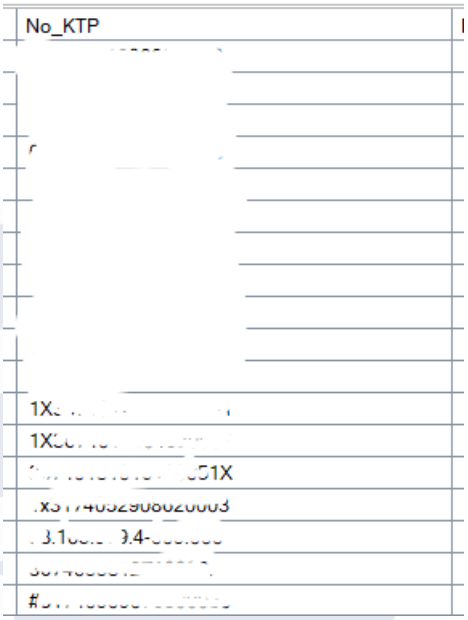
Contoh pola KTP tidak valid yang ditemukan antara lain:

a. 317509XXXX###
b. 3517-0820-....
c. 3271X9382736
d. dan variasi lainnya

Hal ini menunjukkan adanya *human error* dalam proses input data atau ketidaksesuaian standar format penulisan.

Pada gambar hasil *query* validasi KTP, ditemukan beberapa nomor KTP pelanggan yang tidak memenuhi format standar nasional. Nomor KTP seharusnya terdiri dari 16 digit numerik tanpa huruf maupun simbol, namun hasil yang muncul menunjukkan adanya karakter yang tidak sesuai. Contoh nomor KTP tidak valid yang teridentifikasi antara lain:

a. 1X…
b. 1X… (duplikasi kesalahan pola yang sama)
c. …..X (angka yang diakhiri dengan huruf X)
d. #… (nomor KTP berawalan simbol pagar)



Gambar 3. 16 Hasil KTP yang tidak valid

Simbol seperti # tidak seharusnya muncul pada nomor KTP dan mengindikasikan:

a. data placeholder,
b. kesalahan input manual,
c. atau data rusak (corrupted) saat migrasi sistem.

Hasil ini menegaskan bahwa:

a. Sebagian data KTP masih tidak valid, meskipun sebelumnya banyak data yang sudah MATCH antara MDM dan CX.
b. KTP tidak valid berpotensi mengganggu:
    1) verifikasi identitas pelanggan,
    2) integrasi data antar sistem,
    3) proses analitik berbasis NIK,
    4) proses audit dan kepatuhan (compliance).
c. Kasus seperti huruf "X" menandakan adanya kemungkinan data dummy atau placeholder yang harus segera diperbaiki.
d. Simbol seperti # menunjukkan kesalahan input yang tidak terdeteksi oleh sistem front-end.

Selain melakukan pengecekan nomor KTP,dilakukan juga validasi data nomor telepon (mobile number) antara dua database, yaitu:

a. MDM (*Master Data Management*) – penyimpanan utama data pelanggan
b. OneSmile – sistem aplikasi loyalty pelanggan yang memiliki data input dari pengguna

Tujuan dari proses ini adalah untuk mengetahui pelanggan yang memiliki nomor telepon tercatat di OneSmile, tetapi nomor telepon tersebut kosong di MDM. Kondisi ini penting untuk diidentifikasi, karena MDM seharusnya menjadi sumber data terlengkap, dan ketidaklengkapan pada MDM akan berpengaruh pada proses integrasi dan operasional lainnya.

```
1  SELECT
2      MDM.IDKTP AS KTP,
3      MDM.FullName AS Nama,
4      MDM.IDNPWP15,
5      MDM.IDNPWP16,
6      MDM.MobileNumber AS MDM_Mobile,
7      OSL.member_username AS OneSmile_Mobile
8  FROM SD_100_DE_LH_MDM.dbo.MDM_CUSTOMER_INDIVIDUAL AS MDM
9  INNER JOIN SD_100_DE_LH_ONESMILE.dbo.OSL_member AS OSL
10     ON MDM.IDKTP = OSL.member_str_ktp
11 WHERE
12     (MDM.MobileNumber IS NULL
13      OR LTRIM(RTRIM(MDM.MobileNumber)) = '' )
14     AND (OSL.member_username IS NOT NULL
15          AND LTRIM(RTRIM(OSL.member_username)) <> '');
16
```

Gambar 3. 17 Query Perbandingan Mobile

Gambar tersebut menampilkan *query* SQL yang digunakan untuk mencari pelanggan dengan kondisi:

a. Mobile number di MDM kosong, tetapi
b. Mobile number di OneSmile terisi

```
SELECT
  MDM.IDKTP AS KTP,
  MDM.FullName AS Nama,
  MDM.IDNPWP15,
  MDM.IDNPWP16,
  MDM.MobileNumber AS MDM_Mobile,
  OSL.member_username AS OneSmile_Mobile
```

Bagian ini menampilkan:

a. IDKTP → sebagai identifier pelanggan
b. Nama pelanggan

c. NPWP (15 dan 16 digit) bila tersedia
d. Nomor telepon dari MDM
e. Nomor telepon dari OneSmile

Dengan ini, tim dapat melakukan pengecekan data lintas sistem secara langsung.

```
FROM SD_100_DE_LH_MDM.dbo.MDM_CUSTOMER_INDIVIDUAL AS MDM
INNER JOIN SD_100_DE_LH_ONESMILE.dbo.OSL_member AS OSL
ON MDM.IDKTP = OSL.member_str_ktp
```

a. Kedua tabel di-*join* menggunakan ID KTP sebagai penghubung.
b. Hal ini memastikan bahwa data yang dibandingkan berasal dari pelanggan yang sama.

```
WHERE
  (MDM.MobileNumber IS NULL
  OR LTRIM(RTRIM(MDM.MobileNumber)) = '')
```

Bagian ini menyeleksi pelanggan yang:

a. Tidak memiliki nomor telepon di MDM (NULL)
b. Atau *mobile number* berisi string kosong (''), biasanya akibat kesalahan input

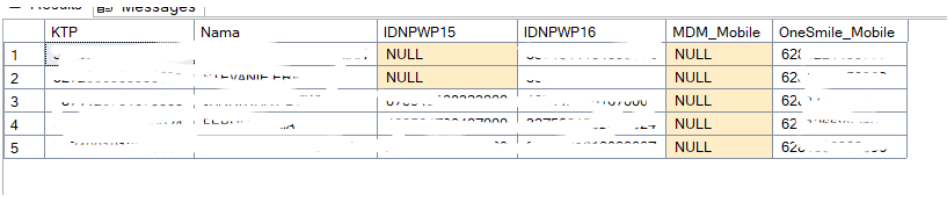
Jika kondisi ini terpenuhi, pelanggan dianggap tidak memiliki data mobile di MDM.

```
AND
  (OSL.member_username IS NOT NULL
  AND LTRIM(RTRIM(OSL.member_username)) <> '')
```

Bagian ini memastikan bahwa:

a. Mobile number ada dan terisi di OneSmile
b. Bukan string kosong
c. Sudah melewati trimming untuk menghapus spasi berlebih

Artinya, pelanggan memiliki mobile number yang valid di sistem OneSmile, tetapi tidak tercatat di MDM.



	KTP	Nama	IDNPWP15	IDNPWP16	MDM_Mobile	OneSmile_Mobile
1			NULL		NULL	62(
2			NULL		NULL	62
3					NULL	62
4					NULL	62
5					NULL	62

Gambar 3. 18 Hasil Perbandingan Mobile

Gambar menunjukkan hasil dari eksekusi *query* perbandingan nomor telepon antara database MDM dan OneSmile, di mana fokus pencarian adalah pelanggan yang tidak memiliki nomor telepon di MDM, tetapi memiliki nomor telepon terdaftar di OneSmile. Pada gambar tersebut terlihat beberapa baris data yang menampilkan kombinasi informasi KTP, nama pelanggan, NPWP, serta kondisi nomor telepon pada kedua sistem. Secara umum, pola yang muncul menunjukkan bahwa:

a. Nomor telepon pada MDM kosong (NULL)
b. Nomor telepon pada OneSmile terisi

1. Kolom KTP (IDKTP)

a. 345
b. 12

Nilai KTP yang muncul pada gambar digunakan sebagai penghubung antara kedua sistem—meskipun contoh yang tampil tampak bukan format KTP 16 digit, ini merupakan data ilustratif dari hasil *query*. Nilai KTP tetap digunakan sebagai acuan untuk mengidentifikasi customer yang sama pada kedua database.

2. Kolom Nama (FullName)

a. VANIT
b. ILLUR.

Nama pelanggan diambil dari database MDM, yang menunjukkan bahwa identitas pelanggan sudah tercatat meskipun nomor teleponnya masih kosong.

3. Kolom NPWP (IDNPWP15 dan IDNPWP16)

Nilai pada kolom NPWP terlihat **NULL**, menandakan bahwa pelanggan belum memiliki data NPWP atau sistem belum mencatatnya.

Hal ini tidak mempengaruhi analisis nomor telepon, tetapi memberikan gambaran bahwa profil pelanggan belum sepenuhnya lengkap.

4. MDM_Mobile

Pada gambar, seluruh nilai MDM_Mobile adalah **NULL**, menunjukkan bahwa:

a. Tidak ada nomor telepon yang tersimpan di sistem MDM
b. Data pelanggan di MDM belum terupdate
c. Terjadi ketidakkonsistenan data antara sistem MDM dan OneSmile

Kekosongan ini menjadi akar masalah yang dicari melalui *query*.

5. OneSmile_Mobile

a. 62…
b. 62-
c. 62,
d. 620
e. Dalam beberapa kasus ada prefix **www.** yang jelas tidak valid sebagai nomor telepon

Meskipun MDM tidak memiliki data mobile, OneSmile menunjukkan bahwa:

a. Pelanggan memiliki nomor telepon yang terdaftar
b. Nomor telepon disimpan dengan variasi format, misalnya:
c. menggunakan prefix internasional **62**
d. tambahan simbol seperti koma **(,)** atau dash **(-)**
e. data yang terinput tidak dalam format numerik murni (contoh prefix "www.")

Hal ini menunjukkan bahwa data pada OneSmile ada, tetapi tidak selalu dalam format yang valid.

Dari Gambar 3.18 terlihat beberapa temuan penting:

1. Ketidaksesuaian Data Antar Sistem

a. MDM tidak memiliki data mobile (NULL)
b. OneSmile memiliki data, meskipun beberapa tidak valid

Ini menegaskan adanya ketidakseimbangan kualitas data antara kedua platform.

2. Mobile Number di OneSmile Tidak Distandardisasi

Beberapa nomor memiliki:

a. Simbol (–, , , .)
b. Karakter non-digit
c. Format tidak lengkap
d. Prefiks seperti "www."

Ini menunjukkan kurangnya validasi input di sistem OneSmile.

3. Perlu Data Enrichment ke MDM

Karena MDM adalah sumber data utama (*single source of truth*), maka:

a. Data valid dari OneSmile harus dipindahkan ke MDM
b. Data tidak valid harus diperbaiki atau diverifikasi ulang
c. Diperlukan proses normalisasi mobile number sebelum pengisian ke MDM

4. Potensi Dampak

Ketidaktelitian ini dapat menyebabkan:

a. Kegagalan komunikasi ke pelanggan
b. Gangguan proses CRM dan marketing
c. Ketidakefektifan kampanye OneSmile
d. Error dalam integrasi antar sistem

Pada tugas selanjutnya, diminta untuk melakukan proses ETL (Extract, Transform, Load) menggunakan Python untuk memasukkan data transaksi dari file CSV ke dalam database internal perusahaan. Karena alasan kerahasiaan sistem, proses awal ETL (extracting dari server, pre-processing, serta koneksi awal) tidak dapat ditampilkan secara lengkap. Oleh karena itu, hanya tahap akhir proses ETL, yaitu bagian *Load*, yang dapat diperlihatkan. Tahap *Load* ini merupakan bagian penting dari ETL, karena bertanggung jawab untuk memasukkan data yang telah dibersihkan dan diproses ke dalam tabel target di database.

```python
# ============================
# 4. LOOP SETIAP FILE CSV
# ============================
for csv_path in csv_files:
    df = pd.read_csv(csv_path, dtype=str)
    df = df.fillna("")

    print(f"Total rows loaded from {csv_path}: {len(df)}")

    for index, row in df.iterrows():
        cursor.execute(insert_query,
            row['NPWP_PEMBELI'],
            row['NOMOR_FAKTUR'],
            row['TANGGAL_FAKTUR'],
            row['NAMA_PENGUSAHA'],
            row['NAMA_PEMBELI'],
            row['ALAMAT_PEMBELI'],
            row['NPWP_PENGUSAHA'],
            row['NAMA_FILE']
        )

conn.commit()
cursor.close()
conn.close()

print("✔ Insert all CSV files complete!")

```

Gambar 3. 19 Script ETL Python

1. Melakukan Loop untuk Setiap File CSV

for csv_path in csv_files:

df = pd.read_csv(csv_path, dtype=str)

df = df.fillna("")

print(f"Total rows loaded from {csv_path}: {len(df)}")

Pada tahap ini:

a. Program melakukan loop untuk setiap file CSV yang sudah terdeteksi sebelumnya.
b. Semua kolom dibaca sebagai string (dtype=str) untuk mencegah error format.
c. Nilai kosong seperti NaN diganti menjadi string kosong ("").
d. Program menampilkan jumlah baris pada setiap file sebagai log proses.

Ini memastikan bahwa seluruh file CSV dapat diproses secara konsisten.

2. Looping Setiap Baris Data

for index, row in df.iterrows():

cursor.execute(insert_*query*,

row['NPWP_PEMBELI'],

```
        row['NOMOR_FAKTUR'],
        row['TANGGAL_FAKTUR'],
        row['NAMA_PENGUSAHA'],
        row['NAMA_PEMBELI'],
        row['ALAMAT PEMBELI'],
        row['NPWP_PENGUSAHA'],
        row['NAMA_FILE']
```

Bagian ini adalah inti dari proses *Load*:

a. Program membaca setiap baris dalam dataframe
b. Untuk setiap baris, program menjalankan *query* INSERT
c. Kolom yang dimasukkan meliputi:
    1) NPWP pembeli
    2) Nomor faktur
    3) Tanggal faktur
    4) Nama pengusaha
    5) Nama pembeli
    6) Alamat pembeli
    7) NPWP pengusaha
    8) Nama file sumber

Semua nilai diambil dari dictionary `row[...]` dan dimasukkan ke database melalui cursor.

3. Commit dan Menutup Koneksi

```
conn.commit()
cursor.close()
conn.close()
print("✓ Insert all CSV files complete!")
```

Tahap terakhir:

a. `conn.commit()` memastikan bahwa seluruh perubahan disimpan ke database.
b. Cursor dan koneksi ditutup untuk menghindari kebocoran memori (*resource leak*).
c. Program menampilkan pesan bahwa seluruh proses insert telah selesai.

Langkah ini merupakan penutup proses ETL dan menandakan bahwa seluruh data berhasil dimuat.

3.3.5 Kendala yang Ditemukan

Selama menjalani proses pekerjaan, ditemukan berbagai kendala, baik yang bersifat teknis maupun non-teknis, di setiap tahapan penugasan. Kendala-kendala ini bukan ditujukan untuk menyalahkan pihak manapun, melainkan dijadikan bahan refleksi untuk memberikan gambaran objektif mengenai tantangan yang dihadapi selama proses pembelajaran.

1. Keterbatasan Akses dan Hak Pengguna Sistem: Saat melakukan pengecekan data di sistem MDM, CX, dan OneSmile, Disadari bahwa tidak semua server bisa diakses secara penuh. Beberapa informasi memerlukan izin tambahan, sehingga proses validasi kadang berjalan lebih lambat. Kendala ini dimaklumi karena merupakan bagian dari mekanisme keamanan untuk melindungi data perusahaan.
2. Perbedaan Struktur dan Standar Format Antar Sistem: Dalam proses membandingkan data antara MDM, CX, dan OneSmile, disadari bahwa setiap sistem memiliki struktur dan format penyimpanan data yang berbeda. Contohnya, format KTP dan nomor telepon di CX sudah dinormalisasi, sementara di OneSmile masih terlihat variasi format, seperti penggunaan tanda baca, huruf, atau digit yang tidak lengkap. Perbedaan ini membuat proses penyamaan data (data matching) memerlukan perhatian ekstra dan penyesuaian tambahan.
3. Ketidakkonsistenan Data Historis: Ditemukan sejumlah data lama yang tidak konsisten, misalnya KTP yang memuat huruf "X", simbol, atau karakter non-digit lainnya, serta nomor telepon dengan format yang tidak sesuai standar. Data seperti ini biasanya berasal dari proses input lama atau migrasi sistem sebelumnya. Kondisi ini mengharuskan untuk melakukan pengecekan manual lebih intensif, karena tidak semua

ketidakwajaran dapat terdeteksi otomatis oleh sistem.

4. Waktu Eksekusi *Query* yang Cukup Lama: Beberapa *query*, terutama yang melibatkan tabel besar seperti MDM_CUSTOMER_INDIVIDUAL atau OSL_member, memerlukan waktu eksekusi cukup lama karena volume data yang sangat besar. Kondisi ini kadang memperlambat proses analisis dan menuntut penyesuaian strategi, misalnya menggunakan sub*query*, indexing yang tepat, atau membagi data menjadi batch. Selain itu, waktu eksekusi yang panjang membuat debugging menjadi lebih menantang karena setiap perubahan pada *query* harus diuji ulang dari awal. Situasi ini memberikan pengalaman langsung mengenai pentingnya merancang *query* yang efisien di lingkungan data berskala besar.
5. Kesalahan Format Saat Melakukan ETL Python: Dalam proses ETL menggunakan Python, beberapa kali ditemukan kendala, seperti perbedaan pemetaan kolom, karakter tak terduga dalam file CSV, atau baris kosong yang membuat script gagal dijalankan. Beberapa file CSV juga memiliki format tanggal atau NPWP yang tidak konsisten, sehingga perlu dilakukan pembersihan data tambahan sebelum proses loading ke database.
6. Keterbatasan Dokumentasi Dataset: Tidak semua dataset memiliki dokumentasi lengkap terkait arti kolom, asal data, atau hubungan antar tabel. Hal ini membuat perlu adanya konsultasi dengan mentor untuk memastikan interpretasi data sesuai konteks. Selain itu juga harus mengeksplorasi isi tabel secara mandiri untuk memahami pola dan struktur data. Pengalaman ini menekankan pentingnya dokumentasi teknis yang baik dalam sebuah organisasi data dan menjadi pelajaran berharga tentang perlunya standar dokumentasi yang konsisten.

7. Perbedaan Validasi Input Antar Sistem: Di tiap sistem, MDM, CX, dan OneSmile, cara validasi data berbeda-beda, sehingga hasil input tidak selalu konsisten. Beberapa sistem membiarkan karakter tertentu, seperti simbol pada nomor telepon, sehingga perlu ditambahkan filter agar analisis data tetap akurat. Perbedaan ini kadang menimbulkan ketidaksesuaian yang harus dicek secara manual. Diperlukan pemahaman karakter masing-masing sistem untuk menetapkan aturan validasi yang tepat. Situasi ini menekankan pentingnya menyelaraskan aturan input agar kualitas data tetap terjaga.
8. Adaptasi terhadap Tools dan Lingkungan Kerja: Sebagai peserta program, harus cepat menyesuaikan diri dengan berbagai tools seperti SQL Server, Python ETL, Power BI, dan platform internal lain. Adaptasi ini membutuhkan waktu, terutama untuk memahami alur data dan dependensi antar tabel di beberapa sistem sekaligus. Namun, pengalaman ini menjadi kesempatan berharga untuk meningkatkan keterampilan teknis secara cepat. Interaksi langsung dengan mentor dan tim juga membantu mempercepat pemahaman tentang cara kerja tool dan proses operasional perusahaan. Secara keseluruhan, pengalaman ini meningkatkan kemampuan dalam bekerja di lingkungan data yang kompleks dan dinamis.

3.3.6 Solusi atas Kendala yang Ditemukan

Sebagai tindak lanjut dari berbagai kendala yang dihadapi selama menjalani proses pekerjaan, diterapkan sejumlah solusi yang bersifat praktis dan konstruktif. Solusi-solusi ini tidak hanya membantu menyelesaikan permasalahan saat itu, tetapi juga memberikan pemahaman lebih dalam mengenai proses pengelolaan data perusahaan serta cara menghadapi perbedaan struktur sistem.

1. Meminta Akses dan Bimbingan kepada Mentor: Untuk mengatasi keterbatasan akses ke beberapa server tertentu, dilakukan koordinasi dengan mentor dan tim terkait. Diajukan permintaan akses tambahan sesuai kebutuhan analisis, atau meminta bantuan mentor untuk melakukan pengecekan tertentu yang hanya dapat dilakukan oleh pihak dengan akses penuh. Langkah ini membantu menjaga keamanan data sekaligus tetap memungkinkan analisis berjalan dengan baik.
2. Menyesuaikan Format Data dan Membuat Normalisasi Tambahan: Perbedaan format antar sistem seperti MDM, CX, dan OneSmile diatasi dengan melakukan normalisasi data menggunakan SQL maupun Python. Misalnya:
    a. Membersihkan simbol pada nomor telepon: Membersihkan simbol pada nomor telepon dilakukan karena beberapa sistem menyimpan data dengan tanda baca seperti "-", "/", atau tanda kurung. Simbol-simbol tersebut dapat mengganggu proses validasi dan perbandingan data antar sistem. Dengan menggunakan fungsi replace pada SQL atau Python, nomor telepon dapat dinormalisasi menjadi deretan angka murni. Proses ini memastikan bahwa data telepon siap digunakan untuk analisis dan pengecekan kualitas data.
    b. Menghapus spasi, tanda baca, dan karakter non-digit: Penghapusan spasi, tanda baca, serta karakter non-digit dilakukan untuk menghindari kesalahan saat melakukan join atau matching data. Langkah ini penting karena format penyimpanan antar sistem tidak selalu seragam. Dengan normalisasi tersebut, dapat dipastikan bahwa nilai yang dibandingkan berasal dari pola data yang sama. Upaya ini meningkatkan akurasi hasil analisis dan mencegah data dianggap mismatch akibat perbedaan format.

c. Menyamakan format KTP sebelum proses matching: Menyamakan format KTP dilakukan dengan cara menghapus karakter tambahan dan memastikan jumlah digit sesuai standar nasional. Tindakan ini membantu mengatasi data KTP yang sudah dinormalisasi di satu sistem namun belum di sistem lainnya. Dengan menyatukan format, proses matching antara MDM, CX, dan OneSmile dapat dilakukan secara lebih konsisten. Hasil normalisasi ini terbukti penting untuk mengurangi risiko kesalahan klasifikasi MATCH, NOT MATCH, maupun INCOMPLETE DATA.

3. Melakukan Validasi Manual untuk Data Tidak Lazim: Untuk data historis yang tidak konsisten seperti KTP dengan huruf "X", tanda pagar, atau simbol lain, dilakukan verifikasi manual dan mencatat pola kesalahan. Dengan cara ini, dapat disusun filter tambahan agar kesalahan serupa dapat terdeteksi otomatis pada *query* berikutnya.
4. Mengoptimalkan *Query* dan Menggunakan Teknik Pemrosesan Bertahap: Waktu eksekusi *query* yang panjang diatasi dengan:
    a. Menggunakan sub*query* untuk mengurangi jumlah data awal
    b. Melakukan trimming dan normalisasi dalam satu proses
    c. Menjalankan *query* secara bertahap jika data terlalu besar
    d. Menggunakan filtering bertahap untuk menghindari full-table scan
5. Memperbaiki dan Menyesuaikan Script ETL Python: Untuk mengatasi error saat proses ETL Python:
    a. Menambahkan pengecekan format pada kolom tertentu
    b. Menyertakan try–except untuk meminimalkan crash

saat ada anomali data

c. Mengonversi semua kolom menjadi string untuk menghindari kesalahan tipe data
d. Melakukan cleaning data sebelum load

6. Mengkonsultasikan Interpretasi Dataset kepada Mentor: Ketika menemukan dataset tanpa dokumentasi atau nama kolom yang ambigu, dengan melakukan diskusi dengan mentor untuk memastikan pemahaman yang benar. Pendekatan ini membantu untuk menghindari kesalahan analisis dan memastikan bahwa setiap langkah yang diambil sesuai dengan alur kerja yang berlaku.
7. Menambahkan Validasi Input pada Tahap Analisis: Perbedaan validasi antar sistem seperti KTP dan mobile number ditangani dengan membuat aturan filter tambahan pada *query* SQL, seperti:
    a. Pengecekan karakter non-digit: Pengecekan karakter non-digit dilakukan untuk memastikan bahwa kolom KTP dan mobile number hanya berisi angka. Langkah ini penting karena beberapa sistem tidak membatasi input, sehingga karakter seperti huruf atau simbol dapat masuk ke database. Dalam menambahkan filter pada *query*, dapat diidentifikasi baris yang tidak memenuhi standar format numerik. Proses ini membantu menjaga integritas data sebelum dilakukan analisis lebih lanjut.
    b. Pembersihan simbol dan huruf: Pembersihan simbol dan huruf dilakukan untuk mengatasi perbedaan format input antar sistem yang terkadang menyertakan tanda baca atau karakter lain. Proses ini dilakukan dengan fungsi SQL untuk menghapus karakter yang tidak diinginkan, sehingga data menjadi seragam. Normalisasi ini memastikan bahwa perbandingan antar

sistem dapat dilakukan secara akurat. Hasilnya, data KTP dan mobile number menjadi lebih konsisten dan siap diproses.

c. Deteksi mobile number yang tidak valid: Deteksi mobile number yang tidak valid dilakukan untuk memisahkan nomor yang tidak memenuhi aturan standar, seperti panjang digit yang salah atau awalan yang tidak sesuai. Validasi ini membantu mengidentifikasi data yang perlu. Proses analisis menjadi lebih efisien dan akurat. Langkah ini juga turut mendukung peningkatan kualitas data pelanggan secara keseluruhan.

d. Pengeluaran data paspor agar tidak salah dikategorikan: Pengeluaran data paspor dilakukan karena beberapa pelanggan menggunakan paspor sebagai identitas alternatif, yang formatnya berbeda dari KTP. Jika tidak dipisahkan, data paspor berpotensi dianggap sebagai data KTP yang tidak valid. Untuk menghindari kesalahan tersebut, dapat ditambabhkan kondisi pada *query* agar paspor otomatis dikeluarkan dari analisis KTP.

8. Adaptasi Tools Melalui Pembelajaran Mandiri: Untuk menyesuaikan diri dengan berbagai tools seperti SQL Server, OneSmile DB, dan Python ETL, dapat melakukan pembelajaran mandiri melalui dokumentasi resmi.