

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Selama proses pengerjaan tugas-tugas, peserta magang selalu mendapatkan bimbingan dan arahan langsung dari mentor serta rekan kerja pada Tim Customer Campaign Fraud (CCF). Adapun *flow* kedudukan peserta magang selama bekerja di PT Bank Central Asia Tbk sebagai berikut.

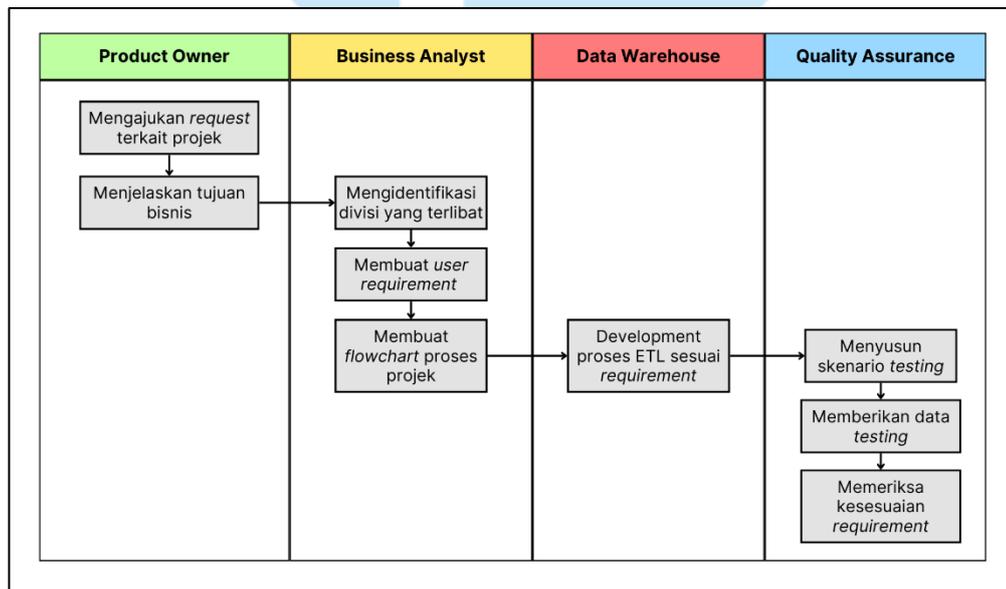


Gambar 3.1 Kedudukan Peserta Magang di BCA

Pada Gambar 3.1, terdapat kedudukan peserta magang selama bekerja di BCA. Peserta magang berada di bawah naungan unit kerja Data Management B, khususnya pada Tim Historical Development dan pada sub tim Customer Campaign Fraud. Data Management B sendiri dikepalai oleh Bapak Suwandi yang mengawasi seluruh tim data, termasuk Tim Historical Development. Sementara itu, untuk Tim Historical Development dikepalai oleh Ibu Seluya yang bertanggung jawab atas

semua aktivitas sub tim dalam pengelolaan data historis di Data Warehouse BCA. Kemudian, adapun mentor atau *team leader* dari Tim Customer Campaign Fraud, yaitu Kakak Fanny Suhendra yang juga menjadi ketua sub tim ini. Kak Fanny menjadi mentor dan pembimbing untuk setiap proyek yang dikerjakan oleh peserta magang. Mentor selalu melaporkan hasil pekerjaan Tim CCF kepada Kepala Historical Development.

Tim Customer Campaign Fraud (CCF) sendiri beranggotakan lima orang, termasuk mentor dan peserta magang. Dari total anggota tersebut, tiga orang ditempatkan di kantor pusat Menara BCA sedangkan dua lainnya ditempatkan di Wisma BCA Foresta (Bumi Serpong Damai). Dalam proses pengerjaan tugas dan proyek, peserta magang terus berkoordinasi secara langsung dengan mentor dan satu rekan kerja. Tim CCF selalu mengadakan *meeting update progress* proyek yang dilaksanakan setiap hari Senin melalui *platform* Microsoft Teams.



**Gambar 3.2** Koordinasi dengan Divisi Lain

Selain berkoordinasi dengan tim CCF sendiri, peserta magang juga berkoordinasi dengan berbagai tim di divisi lain untuk melaksanakan proyeknya sesuai pada Gambar 3.2. Biasanya, proses dimulai dari Product Owner (PO) yang mengajukan *request* untuk terciptanya sebuah tujuan bisnis tertentu, kemudian

terbentuklah proyek untuk mencapai tujuan tersebut. PO juga menjelaskan ke divisi terlibat terkait tujuan bisnis yang ingin dicapai. Selanjutnya, pihak Business Analyst (BA) akan menentukan pihak-pihak yang terlibat dalam proyek, menyusun *user requirement* kepada pihak-pihak terutama DWH, dan membuat diagram alur (*flowchart*) proses proyek yang dilakukan. Setelah *requirement* ditetapkan, dari sisi Tim Data Warehouse siap mengembangkan proses ETL sesuai dengan *requirement* tersebut. Apabila pihak DWH sudah menyelesaikan proses *development*, maka selanjutnya Tim Quality Assurance atau biasa disebut Tim UAT (User Acceptance Testing) akan bertugas menyusun skenario *testing*, memberikan data *testing* selama masa pengujian, dan memeriksa hasil keseluruhan apakah sudah sesuai dengan *requirement* awal. Seluruh kerja sama dan koordinasi ini akan dilakukan melalui Microsoft Teams, di mana terdapat grup khusus untuk setiap proyek sebagai komunikasi antar tim melalui via *chat* maupun *meeting sharing requirement*.

### 3.2 Tugas dan Uraian Kerja Magang

Selama periode kerja magang mahasiswa yang berlangsung kurang lebih lima bulan dari tanggal 15 Januari hingga 19 Juni 2025, peserta magang telah melaksanakan berbagai proyek Tim CCF dengan linimasa pelaksanaan pada tabel berikut:

**Tabel 3.1** Linimasa Proyek Kerja Magang

No	Kegiatan	Tanggal Mulai	Tanggal Selesai
<b>Pengenalan Lingkungan Perusahaan</b>			
1	Pengenalan struktur Tim DTM B dan alur kerjanya	15 Januari	16 Januari
2	Instalasi dan mempelajari <i>tools</i> DWH, serta <i>briefing</i> proyek Tim CCF	17 Januari	24 Januari
<b>Proyek Penalti Nasabah BCA Prioritas-Solitaire</b>			
3	<i>Sharing requirement</i> dan <i>timeline</i> Proyek Penalti Nasabah Prio-Soli	26 Februari	26 Februari

4	Menyiapkan <i>source file</i> dan membuat <i>table-table</i> yang diperlukan di Oracle dan Big Data	27 Februari	27 Februari
5	<i>Development</i> Customer Penalty Temp 1, Temp 2, dan Fact di Informatica PowerCenter.	28 Februari	3 Maret
6	<i>Development offload</i> Customer Penalty Fact di Informatica Developer	4 Maret	4 Maret
7	User Acceptance Testing	5 Maret	13 Maret
8	Persiapan implementasi ke Production	14 Maret	17 Maret
9	Implementasi Proyek Penalti Nasabah Prio-Soli ke Production	20 Maret	20 Maret
<b>Proyek Penurunan Data Log Utilisasi OCR Engine Glair</b>			
10	<i>Sharing knowledge</i> Proyek OCR Engine Glair oleh mentor	6 Maret	6 Maret
11	<i>Development</i> OCR Engine Log Temp 1, Temp 2, dan Fact di Informatica PowerCenter	7 Maret	10 Maret
12	Penambahan kolom <i>table</i> Big Data dan <i>development</i> Informatica Developer	11 Maret	12 Maret
13	User Acceptance Testing	20 Maret	8 April
14	Persiapan implementasi ke Production	14 April	17 April
15	Implementasi Proyek Log OCR Engine Glair ke Production	21 April	21 April
<b>Proyek Enhancement Penambahan Source IE pada Survey Gallup</b>			

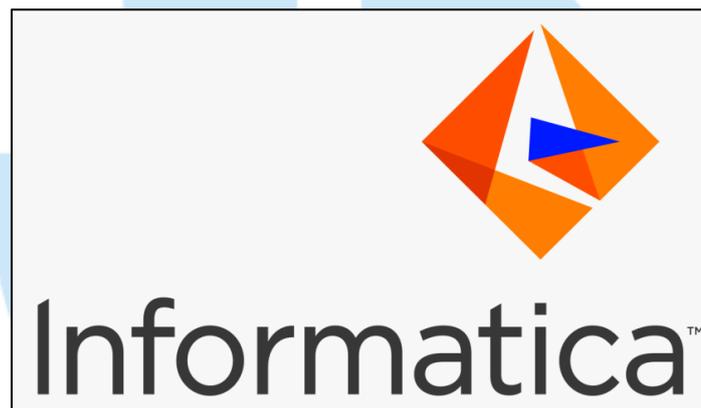
16	<i>Sharing knowledge</i> Proyek Penambahan Source IE oleh rekan tim	18 Maret	18 Maret
17	Menyiapkan <i>source</i> dan membuat <i>table</i> baru di Oracle dan Big Data	18 Maret	18 Maret
18	<i>Development</i> Survey Gallup di Informatica PowerCenter	19 Maret	21 Maret
19	<i>Development</i> Survey Gallup di Informatica Developer	22 Maret	8 April
20	User Acceptance Testing	9 April	11 April
21	Persiapan implementasi ke Production	15 April	17 April
22	Implementasi Proyek Source IE Survey Gallup ke Production	22 April	22 April
<b>Proyek Enhancement Datamart Penurunan CASA Nasabah</b>			
22	<i>Sharing knowledge</i> Proyek Penurunan CASA oleh rekan tim	2 Mei	2 Mei
23	<i>Development table</i> Temp CASA di Informatica PowerCenter	5 Mei	6 Mei
24	<i>Request</i> penurunan data Production ke Development kepada Tim SQI	7 Mei	8 Mei
25	Self Improvement Testing	9 Mei	15 Mei
26	User Acceptance Testing	16 Mei	28 Mei
27	Persiapan implementasi ke Production	2 Juni	12 Juni
28	Implementasi Proyek Enhancement Datamart Penurunan CASA ke Production	17 Juni	17 Juni

Proyek New Pricing WhatsApp			
29	<i>Sharing requirement</i> Proyek New Pricing WhatsApp	26 Maret	17 Maret
30	<i>Development table dan mapping</i> WA Push Notification, WA Conversation, dan WA Campaign History Fact di Informatica PowerCenter	15 Mei	20 Mei
31	<i>Development</i> WA Push Notification, WA Conversation, dan WA Campaign History Fact di Informatica Developer	21 Mei	21 Mei
32	User Acceptance Testing	2 Juni	19 Juni
33	Persiapan implementasi ke Production	19 Juni	24 Juni
34	Implementasi Proyek New Pricing WhatsApp ke Production	1 Juli	1 Juli

Selama mengerjakan proyek, peserta magang secara konsisten membuat dokumentasi *draft changes*. Dokumentasi ini digunakan untuk mencatat setiap perubahan yang dilakukan pada saat proses *development*, seperti perubahan *mapping*, penambahan kolom *table*, perubahan *logic*, dan lain-lain. Dengan adanya *draft changes*, mentor dan tim dapat dengan mudah memantau dan memahami alasan di balik perubahan yang dilakukan. Kemudian, dalam mendukung penyelesaian proyek-proyek selama periode magang, peserta magang pun memanfaatkan berbagai *tools* dan sistem yang tersedia pada Data Warehouse BCA. Dengan *tools* yang terintegrasi, peserta magang mampu mengolah data secara sistematis sesuai kebutuhan *user*. Selama kegiatan magang, peserta magang melakukan pembelajaran terhadap *tools* yang digunakan dalam proses pengelolaan data dan pengembangan *pipeline* ETL. Proses pembelajaran dilakukan secara

mandiri dengan membaca berbagai materi yang tersedia, mengeksplor dokumentasi internal, serta mempelajari contoh-contoh pekerjaan yang telah tersedia pada *repository* Data Warehouse (DWH). Berikut *tools-tools* yang digunakan sebagai pendukung perangkat kerja:

1. Informatica PowerCenter



Gambar 3.3 Informatica PowerCenter

Informatica PowerCenter merupakan salah satu *tools* ETL (Extract, Transform, Load) yang penting untuk mengelola dan mengintegrasikan data dari berbagai sumber ke dalam Data Warehouse berupa hasil Datamart. *Tools* ini dirancang untuk memroses volume data dalam jumlah besar melalui pemrosesan paralel [11]. *PowerCenter* memiliki beberapa komponen utama yang terbagi-bagi menjadi *tools* terpisah, yakni PowerCenter Designer, Workflow, dan Monitor. Designer berfungsi untuk merancang segala proses ETL dengan membuat *mapping* proses dari sumber data hingga menjadi target. Kemudian, PowerCenter Workflow berfungsi untuk mengatur alur eksekusi proses ETL dengan menentukan urutan *session* serta mengeksekusi *mapping* yang telah dibuat pada Designer. Terakhir, Monitoring memungkinkan peserta magang untuk memantau *log* dan status eksekusi apakah *job* berjalan sukses tanpa adanya *error* atau sebaliknya. Informatica PowerCenter pada DWH BCA terkoneksi dengan *database* Oracle.

2. Informatica Developer

Informatica Developer secara fungsi sangatlah mirip dengan Informatica PowerCenter, yakni *tools* untuk proses ETL. Namun, perbedaannya terletak pada fitur *all-in-one* pada Developer, di mana semua komponen, seperti Designer, Workflow, dan Monitoring dilakukan dalam satu kesatuan aplikasi tanpa terpisah layaknya PowerCenter. Selain itu, Informatica Developer memiliki *user interface* yang lebih modern. Pada DWH BCA, Informatica Developer terkoneksi dengan *database* Hive Big Data sehingga proses ETL pada Developer biasanya digunakan untuk kebutuhan *offload* dan *backup* data dari Oracle ke Hive Big Data. Untuk menghubungkan ke Big Data, Developer, menggunakan koneksi Hadoop dengan *runtime* Blaze dan Spark.

### 3. Oracle



Gambar 3.4 Database Oracle

Oracle merupakan salah satu *tools* penting dalam manajemen basis data yang digunakan secara luas dalam berbagai industri di dunia, termasuk perbankan. Dikembangkan oleh Oracle Corporation dengan *paid license* [12]. Oracle Database memberikan RDMS yang andal, kinerja, dan skalabilitas tinggi sehingga mampu menangani data dalam jumlah besar dalam bentuk tabel [13]. Dalam proyek yang dikerjakan, Oracle kerap kali digunakan untuk membuat *query* SQL berupa DDL (Data Definition Language) dan DML (Data Manipulation Language). Oracle Database terintegrasi dengan *tools* Informatica PowerCenter sehingga target *table* hasil transformasi akan disimpan pada *database* Oracle.

### 4. DBeaver



Gambar 3.5 Database DBEaver

DBEaver adalah *tools* RDMS yang mendukung berbagai jenis *database*, termasuk Big Data. Secara sederhana, DBEaver juga digunakan untuk mengeksekusi *query* SQL berupa DDL dan DML pada Big Data dengan koneksi Hadoop. Dalam ekosistem Big Data terdapat dua Apache utama, yaitu Hive Apache dan Impala Apache. Hive biasa digunakan untuk *query* DDL, seperti Create Table dan menyimpan metadata tabel secara permanen (tidak bisa dihapus). Melainkan Impala, biasanya digunakan untuk *query* DML karena memiliki performa eksekusi yang lebih cepat dalam menangani transaksi data. Pada Data Warehouse BCA, hasil target *table* yang diproses menggunakan Informatica Developer akan disimpan pada *database* Big Data. Maka dari itu, BCA memiliki dua tempat penyimpanan data yang berbeda untuk keperluan *backup* dan *housekeeping*.

#### 5. MobaXterm



Gambar 3.6 MobaXterm

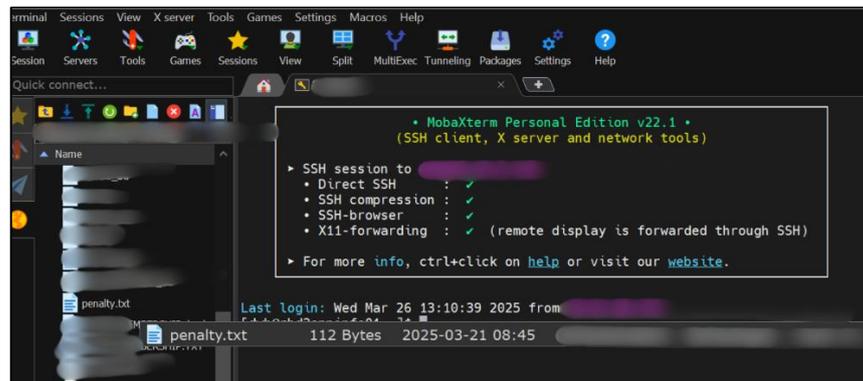
MobaXterm adalah sebuah *tools* terminal berbasis Linux yang sering digunakan untuk mengakses server secara *remote* melalui berbagai protokol, seperti SSH, SFTP, dan lain-lain. Dalam DWH BCA, MobaXterm kerap kali digunakan untuk mengunggah *source file* ke dalam server sehingga dapat diakses oleh Informatica PowerCenter. Selain itu, MobaXterm menjadi tempat untuk mengelola *job-job* berupa *script* dengan format *.sh* yang digunakan untuk menjalankan proses otomatisasi dalam DWH. Terdapat pula file konfigurasi berupa parameter yang menyimpan variabel-variabel penting. MobaXterm memungkinkan penerimaan *file* data dari tim lain yang di-*drop* pada *directory* tertentu.

### 3.2.1 Proyek Penalti Nasabah BCA Prioritas-Solitaire

Dalam memberikan layanan perbankan terbaik untuk nasabahnya, PT Bank Central Asia Tbk membuat dua segmen nasabah eksklusif yang mendapatkan berbagai keuntungan dan fasilitas berbeda dari nasabah biasa, yakni nasabah BCA Prioritas dan BCA Solitaire. Untuk mendapatkan *privilege* tersebut, terdapat ketentuan-ketentuan yang harus dipenuhi oleh nasabah tertentu pada masing-masing segmen. Salah satunya adalah menjaga saldo minimum yang disyaratkan oleh BCA ketika sudah menjadi nasabah BCA Prioritas/Solitaire. Jika nasabah tidak memenuhi dan mempertahankan saldo minimum dalam periode tertentu, maka mereka dapat dikenakan biaya penalti atau biaya layanan tambahan. Oleh karena itu, proyek ini bertujuan untuk *development* tabel pada *database* dan proses ETL yang memungkinkan data penalti nasabah Prioritas-Solitaire tersimpan dan terkelola dalam Data Warehouse sehingga dapat digunakan untuk analisis lebih lanjut.

#### 1. Drop Source File

Sebelum dilakukannya tahap Development, langkah awal yang perlu dilakukan adalah melakukan *drop source file* pada aplikasi MobaXterm.



**Gambar 3.7** Drop Source File Penalty di Server

Terlihat pada Gambar 3.7, *source file* berupa FlatFile .txt telah di-drop ke dalam server DWH menggunakan MobaXterm. Data *source* ini merupakan data *development* yang hanya digunakan untuk keperluan uji coba selama pengembangan berlangsung. Penempatan *file* dilakukan di session DWH pada direktori /monthly karena proses ini termasuk ke dalam *job* bulanan. Nantinya, FlatFile akan digunakan pada *workflow* Informatica melalui parameter direktori *file* tersebut.

## 2. Development

Tahap Development merupakan tahap pengembangan proyek penalti yang dilakukan pada *development environment* DWH sebelum nantinya akan diimplementasikan ke *production*. Dalam proses ini, *user* membutuhkan pembuatan *table* Customer Penalty Fact yang berfungsi sebagai tabel transaksi utama untuk mencatat data penalti nasabah Prioritas-Solitaire. Tabel akan dibuat sesuai dengan *user requirement* pada *database* Oracle dan Big Data. Selanjutnya, proses *mapping* (pemetaan data) akan dilakukan menggunakan dua *tools* utama, yaitu Informatica PowerCenter dan Informatica Developer. Mapping ini mencakup proses pengambilan data dari *source*, transformasi data sesuai *logic* tertentu, hingga *load* hasil transformasi ke tabel Customer Penalty Fact. Berikut tahapan proyek yang dilakukan dari awal hingga akhir secara rinci.

a. Pembuatan Tabel Customer Penalty Fact

Tahap kedua ialah membuat *table-table* yang diperlukan pada proyek ini pada Oracle dan Big Data. Di lingkup *development* Data Warehouse, tabel tersebut sebelumnya belum tersedia sama sekali sehingga perlu dilakukan proses CREATE TABLE dari awal.

```
CREATE TABLE ..._TEMP_..._Y_1(
  ..._NO VARCHAR2(11),
  FLAG_..._P VARCHAR2(1),
  LAST_UPDATE_DT DATE) TABLESPACE ...;

CREATE TABLE ..._TEMP_..._Y_2(
  ROWNUM NUMBER,
  P..._DA_..._K NUMBER,
  ..._NO VARCHAR2(11),
  FLAG_..._P VARCHAR2(1),
  LAST_UPDATE_DT DATE) TABLESPACE ...;

CREATE TABLE ..._FACT(
  PO..._D_..._F NUMBER,
  ..._NO VARCHAR2(11),
  FLAG_..._P VARCHAR2(1),
  LAST_UPDATE_DT DATE) TABLESPACE TLODF;

CREATE INDEX ..._F
ON ..._FACT(PO..._D_..._K, ..._NO ASC) TABLESPACE ...;
```

Gambar 3.8 Query DDL Create Table Penalty Oracle

Pada Gambar 3.8, terdapat *query* SQL untuk Data Definition Language berupa CREATE TABLE pada *database* Oracle. Peserta magang membuat tiga tabel yang diperlukan untuk proyek penalti. Dalam proses pembuatan tabel pada DWH, terdapat ketentuan bahwa sebuah tabel *fact* tidak dapat langsung dibuat dan langsung diisi datanya begitu saja. Namun, setiap pengisian tabel *fact* harus melewati dua *layer* terlebih dahulu yang menyebabkan tabel Customer Penalty Temp 1 dan Customer Penalty Temp 1 perlu dibuat. Tabel *temp* merupakan sebuah tabel *temporary* (sementara) yang akan selalu di-TRUNCATE saat dijalankan pada Informatica PowerCenter agar tidak menyimpan data lama dan hanya memuat data terbaru. Melainkan tabel *fact* yang tidak akan di-TRUNCATE karena berperan sebagai tempat penyimpanan akhir data historis. Pada dasarnya, ketiga tabel memiliki kolom-kolom yang pasti sama, seperti nomor nasabah dan penanda (*flag*) *memberhsip*

sesuai dengan *requirement*. Kemudian, setiap tabel *temp* dan *fact* wajib memiliki kolom `LAST_UPDATE_DT` untuk mencatat tanggal terakhir data masuk ke dalam tabel. Terakhir, dilakukan `CREATE INDEX` pada tabel *fact* penalti berdasarkan *date key* untuk meningkatkan performa eksekusi *query*, terutama saat penyaringan, pencarian, ataupun *join*. Ketiga tabel masing-masing ini akan menjadi *source* dan *target* pada *mapping* Informatica PowerCenter.

```
CREATE EXTERNAL TABLE raw. (
  _NO STRING,
  FLAG_ STRING,
  LAST_UPDATE_DT TIMESTAMP)
PARTITIONED BY (INSERTED_DT STRING)
CLUSTERED BY (_NO) INTO 1 BUCKETS
STORED AS PARQUET
LOCATION 'raw/'
TBLPROPERTIES("parquet.compression"="SNAPPY");

CREATE EXTERNAL TABLE _FACT(
  _NO STRING,
  FLAG_ STRING,
  LAST_UPDATE_DT TIMESTAMP)
PARTITIONED BY (DATE_KEY INT)
CLUSTERED BY (_NO) INTO 1 BUCKETS
STORED AS PARQUET
LOCATION 'fact/'
TBLPROPERTIES("parquet.compression"="SNAPPY");
```

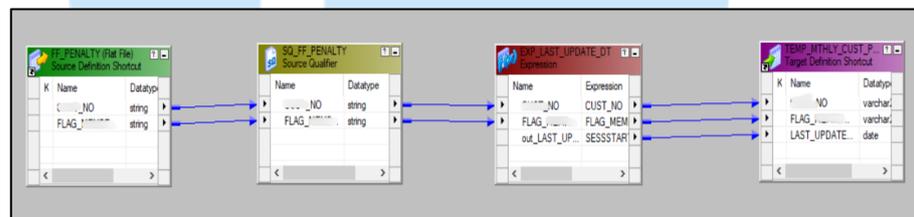
Gambar 3.9 Query DDL Create Table Penalty Big Data

Gambar 3.9 menunjukkan *query* SQL pembuatan tabel pada Hive Big Data. Pada *environment* Big Data, pembuatan tabel proyek penalti dilakukan dengan dua jenis tabel utama, yaitu *raw* dan *fact*. Tabel *raw* bertujuan untuk menyimpan data mentah yang nantinya diambil dari dokumen *flatfile*. Sementara tabel *fact* pada Big Data hanya akan di-*offload* dari tabel *fact* yang sudah ada di Oracle sehingga tidak perlu melalui proses *staging*, seperti di DWH Oracle. Kedua tabel tersebut akan dibuat sebagai EXTERNAL TABLE yang menandakan bahwa struktur tabel ini akan tetap ada meskipun datanya dihapus. Terdapat pula perbedaan sintaks dan struktur pada Hive Big Data, seperti tipe data VARCHAR diganti menjadi STRING dan DATE menjadi TIMESTAMP. Data dalam tabel *raw* akan dipartisi berdasarkan tanggal masuknya data sedangkan *fact* akan dipartisi berdasarkan *date key* yang

mempermudah dalam pencarian dan pengelolaan data. Tabel disimpan di direktori tertentu dalam server, di mana tabel *raw* berada di *path directory /raw* dan tabel *fact* di */fact* dengan format PARQUET.

b. Mapping Informatica PowerCenter

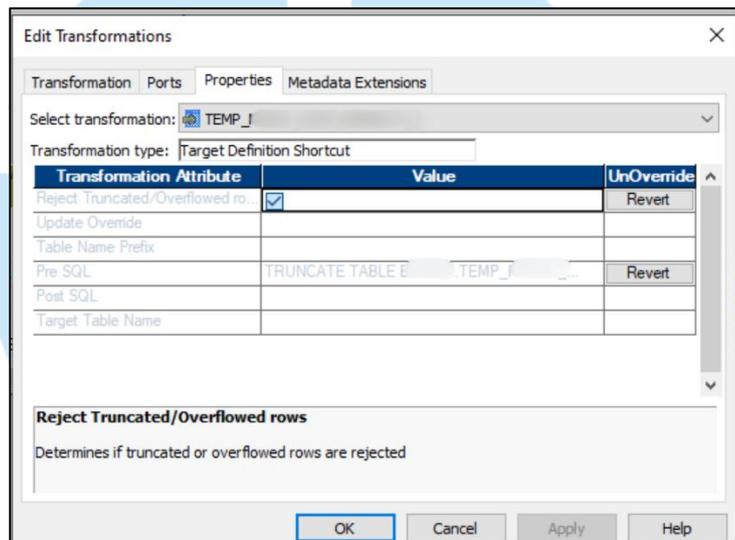
Proses pemetaan atau *mapping* dilakukan menggunakan fitur Designer pada Informatica PowerCenter. Dalam proyek ini, akan dibuat tiga *mapping* utama di Informatica PowerCenter dengan target tabel Oracle dan dua *mapping* di Informatica Developer dengan target tabel Big Data.



Gambar 3.10 Mapping Customer Penalty Temp 1

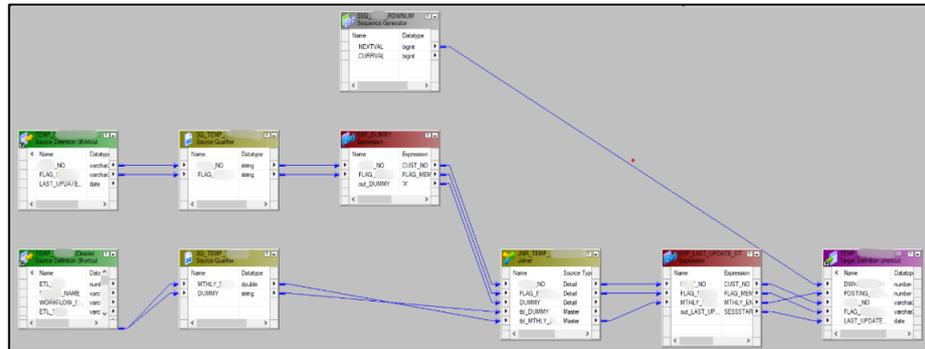
Setelah tabel sudah dibuat pada Oracle dan Big Data, selanjutnya ialah membuat pemetaan data (*mapping*) untuk *temporary* tabel *fact* penalti pada *tools* Informatica PowerCenter bernama Customer Penalty Temp 1, seperti pada Gambar 3.10. Mapping ini akan menjadi *layer* pertama dalam alur pemrosesan data penalti. Mula-mula, peserta magang melakukan *import* Source File berupa *flatfile* penalti.txt ke dalam *repository* DWH BCA. Data mentah ini akan dimasukkan ke dalam Source Qualifier untuk diseleksi data dan kolom-kolom yang digunakan lebih lanjut. Setelah melalui SQ, data dialirkan ke transformasi Expression, di mana pada transformasi ini akan ditambahkan sebuah kolom baru untuk *last update date* yang di-*hardcode* menggunakan *syntax* SESSSTARTTIME. Hal tersebut berfungsi untuk menandakan dan merekam waktu ketika *session* dijalankan sehingga data baru yang masuk akan selalu memiliki

*timestamp* terbaru. Formatnya ialah tanggal-bulan-tahun jam:menit:detik hingga milidetik. Terakhir, data dimasukkan ke dalam Target Definition berupa tabel sementara (*temporary*) bernama Customer Penalty Temp 1 yang sebelumnya telah dibuat pada Oracle. Pada Temp 1, biasanya belum terdapat *logic-logic* tertentu.



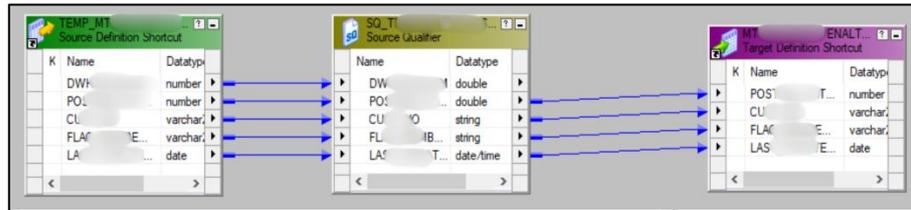
Gambar 3.11 Truncate Table

Pada Gambar 3.11 juga terlihat adanya pengaturan di bagian transformasi Target Definition, di mana untuk seluruh *mapping* tabel *temporary* harus disisipkan sebuah Pre SQL berupa TRUNCATE TABLE. Pengaturan ini dilakukan supaya setiap kali *session* dijalankan, maka isi tabel *temporary* akan dikosongkan terlebih dahulu sebelum dimuat ulang dengan data baru. Data duplikat dan tumpang tindih saat proses ETL berlangsung juga dapat terhindarkan.



Gambar 3.12 Mapping Customer Penalty Temp 2

Mapping kedua pada Gambar 3.12, dibuat untuk menghasilkan tabel Customer Penalty Temp 2 dengan *source* utamanya berasal dari target *table* Customer Penalty Temp 1 yang dihasilkan pada *mapping* sebelumnya. Selain itu, *mapping* ini juga menggunakan *source* tambahan TEMP\_ETL\_XXX berupa tabel *existing* pada DWH, di mana *source* telah disisipkan sebuah Pre SQL WHERE dengan filter *key* = 20 sebagai identifikasi khusus untuk proyek penalti ini. Pada *temp 2*, sudah mulai diterapkan *logic-logic* pemrosesan data. Tujuan menggunakan *source* TEMP\_ETL\_XXX, ialah untuk mendapatkan kolom POSTING\_XXX melalui proses *join*. Tanggal *posting* merupakan Sebelum dilakukan transformasi Joiner, Source Qualifier *temp 1* akan dialirkan ke transformasi Expression terlebih dahulu untuk dilakukan *hardcode* kolom baru DUMMY dengan *value* String 'X'. Kolom DUMMY ini juga terdapat pada TEMP\_ETL\_XXX sehingga dapat dilakukan transformasi Joiner dengan tipe *join* berupa Normal Join menggunakan kondisi kolom DUMMY tersebut sebagai *key*. Setelah itu, data dialirkan ke Expressin kembali untuk menghasilkan *last update* dengan *syntax* SESSSTARTTIME. Terakhir, terdapat pula transformasi Sequence Generator yang berfungsi menghasilkan kolom nomor unik sebagai identifikasi baris dalam DWH. *Mapping* ditutup dengan memasukkan Target Definition, yakni tabel Customer Penalty Temp 2 yang telah dibuat pada Oracle.

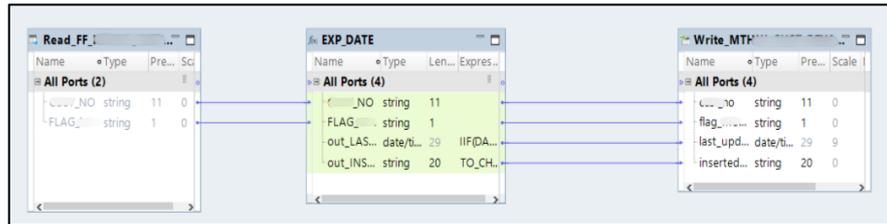


Gambar 3.13 Mapping Customer Penalty Fact

Mapping tabel Customer Penalty Fact pada Gambar 3.13 dimulai dengan menjadikan tabel Customer Penalty Temp 2 sebagai *source* di *mapping* ini. Melalui Source Qualifier, tidak semua kolom dari tabel *temp 2* diambil, melainkan kolom-kolom yang dibutuhkan saja sesuai dengan struktur tabel *fact* yang telah dibuat. Hal ini seperti kolom DWHROWNUM tidak dibutuhkan pada tabel *fact*. Selanjutnya, data langsung dialirkan ke transformasi Target Definition berupa tabel Customer Penalty Fact. Pengaturan target untuk tabel *fact* tidak memerlukan perintah Pre SQL TRUNCATE TABLE karena data bersifat historikal dan akan terus bertambah (bukan di-*replace*). Maka dari itu, kolom akhir dari tabel ini ialah POSTING DATE KEY, XXX\_NO, FLAG XXX, dan LAST UPDATE DT. Pembuatan *mapping* Customer Penalty Fact pada Designer sudah selesai, selanjutnya perlu membuat *mapping* untuk *offload* ke Big Data.

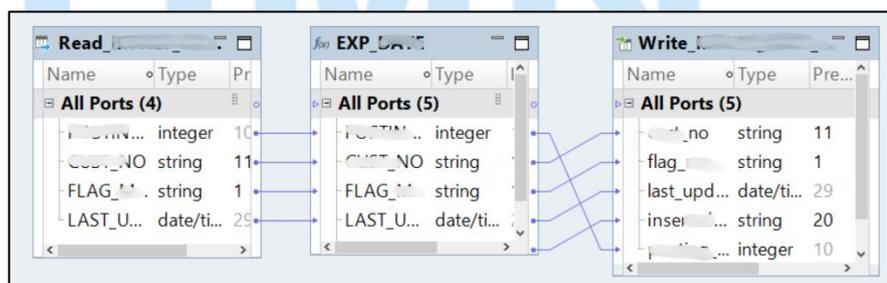
c. Mapping Informatica Developer

Mapping pada Informatica Developer memiliki tujuan utama untuk memindahkan/*offload* tabel target pada *database* Oracle yang sudah diproses melalui PowerCenter Designer ke *environment* Big Data. Disini akan terdapat dua *mapping* yang dibuat, yaitu *mapping* Table Raw dan Table Fact.



Gambar 3.14 Mapping Customer Penalty Raw Big Data

Gambar 3.14 menunjukkan *mapping* untuk table *raw* penalti pada *environment* Big Data melalui Informatica Developer. *Mapping* ini bersifat mentah (*raw*) karena seluruh isi *flatfile* penalti akan dimuat langsung ke tabel target tanpa adanya *logic* kompleks. *File* yang sebelumnya sudah diimpor ke *repository* akan dibaca menggunakan transformasi Read yang kemudian diarahkan ke transformasi Expression untuk menghasilkan dua kolom, yakni tanggal *last update* dan *inserted*. Sesuai ketentuan standar DWH Big Data, kolom tanggal *inserted* wajib ada dan menggunakan format INT sebagai penanda tanggal data masuk ke dalam tabel. Setelah itu, data langsung dikirim ke target melalui transformasi Write ke tabel mentah penalti. Eksekusi *mapping* ini dijalankan menggunakan koneksi berbasis Spark pada Hadoop Big Data.



Gambar 3.15 Mapping Customer Penalty Fact Big Data

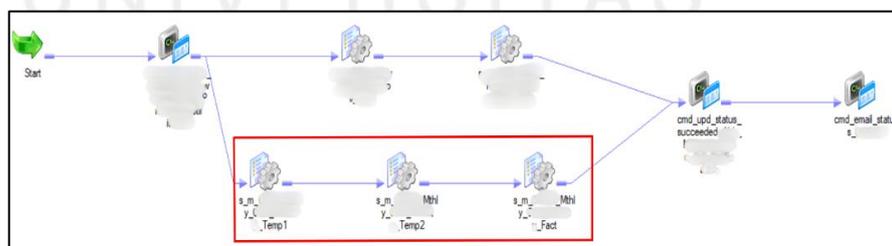
Selanjutnya, perlu dilakukan proses *offload* tabel *fact* penalti ke dari Oracle ke Big Data sehingga tabel juga tersedia pada *environment* ini sesuai Gambar 3.15. Proses dimulai dengan melakukan impor *source* tabel *fact* hasil akhir yang telah diproses sebelumnya di

PowerCenter Designer. *Source* melalui Read akan diarahkan ke transformasi Expression hanya untuk menambahkan kolom dan *logic* tanggal *inserted* yang memberikan informasi waktu kapan data dimuat melalui partisi. Terakhir, data dialirkan ke target tabel *fact* yang terdapat pada Hive Big Data. Eksekusi *mapping* juga menggunakan koneksi Spark Hadoop agar proses *offload* berjalan optimal pada lingkungan Big Data.

### 3. Testing Development

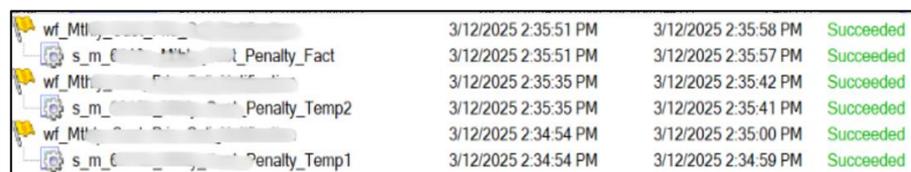
Apabila *development mapping* pada Informatica sudah selesai, maka selanjutnya perlu dilakukan uji coba terhadap *mapping* tersebut. Uji coba ini wajib berupa SIT (Self Improvement Testing) dan UAT (User Acceptance Testing). *Testing* biasanya dilakukan dengan menjalankan *mapping* melalui *workflow*. Pada Informatica PowerCenter, *workflow* bersifat terpisah dan berada pada aplikasi PowerCenter Workflow yang berbeda dari PowerCenter Designer, sedangkan pada Informatica Developer, *workflow* berada pada satu aplikasi yang sama dengan *mapping*. *Workflow* sendiri terdiri dari rangkaian *session* atau *mapping* yang disusun sedemikian rupa agar eksekusinya berjalan secara berurutan sesuai dependensi datanya. Proses *testing* bertujuan untuk memastikan bahwa hasil *development* yang dilakukan peserta magang sudah sesuai dengan kebutuhan dan *requirement* dari *user* terkait, serta memvalidasi hasil akhir tabel *datamart* sudah benar.

#### a. Workflow Informatica PowerCenter



**Gambar 3.16** Workflow Penalti Prio-Soli PowerCenter

Tahap uji coba pada *mapping* Informatica PowerCenter akan dilakukan dengan cara *running task* melalui sebuah *workflow*. Pada proyek penalti Prio-Soli ini, peserta magang menambahkan serangkaian *job* dengan memodifikasi dan menggunakan *workflow* yang sudah ada sebelumnya pada DWH (*existing*), seperti Gambar 3.16. *Workflow* tersebut dimodifikasi dengan menambahkan tiga *session* atau *mapping* (ditandai kotak merah), yakni *temp 1*, *temp 2*, dan *fact* secara paralel di bawah *session* proyek lain yang masih dalam satu rumpun Prioritas-Solitaire. Alur selalu diawali dengan komponen Start, dilanjutkan dengan Command berupa *update* status saat *workflow* dijalankan. Setelah itu, alur akan dipecah menjadi dua cabang, yaitu ke arah *job* penalti dan juga ke arah *job* lainnya. Ketiga *session* yang ditambahkan memiliki alur eksekusi yang saling bergantung satu sama lain karena membutuhkan hasil dari *session* sebelumnya. Kemudian, terdapat Command kembali untuk memberikan informasi status apakah seluruh *job* sukses dijalankan atau terdapat *error*.



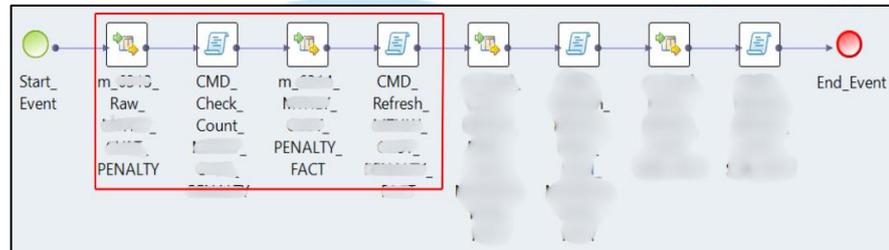
wf_Mit...	3/12/2025 2:35:51 PM	3/12/2025 2:35:58 PM	Succeeded
s_m_..._Penalty_Fact	3/12/2025 2:35:51 PM	3/12/2025 2:35:57 PM	Succeeded
wf_Mit...	3/12/2025 2:35:35 PM	3/12/2025 2:35:42 PM	Succeeded
s_m_..._Penalty_Temp2	3/12/2025 2:35:35 PM	3/12/2025 2:35:41 PM	Succeeded
wf_Mit...	3/12/2025 2:34:54 PM	3/12/2025 2:35:00 PM	Succeeded
s_m_..._Penalty_Temp1	3/12/2025 2:34:54 PM	3/12/2025 2:34:59 PM	Succeeded

**Gambar 3.17** Hasil Running Session Penalti Prio-Soli

Ketiga *session* penalti yang telah dimasukkan pada *workflow* akan dijalankan dan dieksekusi untuk melihat hasil *datamart* tersebut. Hasil eksekusi dari masing-masing *session* akan dimonitor melalui aplikasi PowerCenter Monitoring, seperti pada Gambar 3.17. *Tools* ini akan menampilkan status akhir eksekusi berupa Succeeded atau Failed, serta menyediakan *log* detail yang digunakan untuk memeriksa proses apabila terjadi *error*. Seluruh *session* terkait proyek penalti (*temp 1*, *temp 2*, dan *fact*) telah berhasil dijalankan dengan status Succeeded tanpa adanya *error*. Hal ini menunjukkan proses ETL telah berjalan

sesuai *flow* yang dirancang, serta data telah berhasil dimuat ke masing-masing tabel target, khususnya tabel akhir Customer Penalty Fact.

b. Workflow Informatica Developer



**Gambar 3.18** Workflow Penalti Prio-Soli Informatica Developer

Pada Gambar 3.18, peserta magang juga melakukan modifikasi pada *workflow* yang telah ada sebelumnya di DWH melalui Informatica Developer. *Workflow* penalti Big Data dilakukan dengan menambahkan dua *session/mapping* dan dua CMD (Command Task), seperti yang ditandai dengan kotak merah pada gambar di atas. *Session* yang ditambahkan ialah *mapping* untuk *raw* penalti dan *fact* penalti yang mewakili proses awal dan akhir dari *pipeline* data penalti di Big Data. Alur *workflow* dimulai dari Start Event, kemudian langsung diarahkan ke *mapping raw* penalti. *Mapping raw* selalu dipasangkan dengan Command Task bernama CMD Check Count untuk memverifikasi jumlah data yang masuk. Setelah itu, proses *workflow* dilanjutkan ke *mapping fact* penalti yang dipasangkan dengan CMD Refresh untuk memperbarui partisi atau metadata pada tabel Hive setelah selesai dilakukan proses *insert* data. Command Task tersebut berisi sebuah perintah untuk mengeksekusi *script* dalam *path* tertentu beserta dengan parameter-parameternya. Namun, pada proyek ini, eksekusi *mapping* Big Data tidak dilakukan melalui *workflow*, seperti pada Informatica PowerCenter melainkan langsung dijalankan melalui kanvas *mapping* itu sendiri dengan Running Mapping.

c. Hasil Datamart

POSTING_DATE_KEY	CUST_NO	FLAG_MEMBERSHIP	LAST_UPDATE_DT	cust_no	flag_membership	last_update_dt	inserted_dt	posting_date
2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00
2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00
2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00
2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00
2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00
2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00
2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00
2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00
2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	1234567890	Y	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00
2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	1234567890	N	2025-03-20 10:00:00	2025-03-20 10:00:00	2025-03-20 10:00:00

**Gambar 3.19** Hasil Datamart Customer Penalty Fact

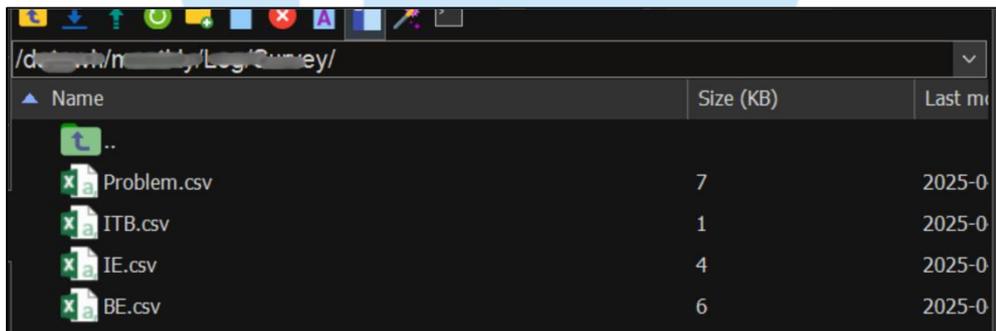
Terakhir, Gambar 3.19 menunjukkan hasil proses data pada tabel *fact* penalti berupa isi *datamart* setelah seluruh *mapping* dijalankan. Sebelah kiri menampilkan data yang berhasil masuk ke tabel *fact* di Oracle sedangkan sebelah kanan menunjukkan hasil yang sama di Big Data. Hasil proses data ini selalu dilaporkan kepada Tim UAT selama masa uji coba (User Acceptance Testing) sebagai bagian dari proses verifikasi kualitas sesuai dengan *user requirement* sebelum diimplementasikan pada lingkungan Production. Implementasi proyek ini dilakukan pada tanggal 20 Maret 2025.

### 3.2.2 Proyek Enhancement Penambahan Source IE pada Survey Gallup

Proyek selanjutnya yang dilaksanakan oleh peserta magang adalah melakukan *enhancement* terhadap proyek Survey Gallup. Proyek ini termasuk proyek *existing* dan sudah berjalan sebelumnya. Namun, kali ini diperlukan sebuah pengembangan untuk menyesuaikan dengan kebutuhan baru dari *user* terkait. Survey Gallup merupakan survei yang dilakukan kepada nasabah BCA yang pernah mengunjungi kantor cabang BCA, mulai dari KCU (Kantor Cabang Utama) hingga KCP (Kantor Cabang Pembantu). Survei memiliki cakupan besar aspek penilaian, seperti kinerja pelayanan, kenyamanan fasilitas, pengalaman bertransaksi, dan lain sebagainya. Data yang dihasilkan dari survei berbentuk verbatim, yaitu komentar-komentar terbuka dari nasabah yang memberikan masukan langsung terkait pengalaman mereka pada kantor cabang BCA tertentu. Data ini akan menjadi sangat penting untuk dikonsumsi oleh tim SPECTA (platform internal BCA untuk aktivitas *campaign*).

Survey Gallup sendiri memiliki beberapa tipe pertanyaan yang dikelompokkan berdasarkan temanya, mulai dari Branch Experience (BE), Branch Relationship (BR), Problem, dan Intent To Buy (ITB). Hal ini membuat proyek Survey Gallup mempunyai banyak *source* berdasarkan tipe-tipe pertanyaan tersebut. Proyek *enhancement* akan dilakukan dengan menghapus *source* data kategori Branch Relationship (BR) yang kemudian digantikan oleh *source* baru bernama Interaction Experience (IE). Pergantian *source* ini bertujuan untuk menyesuaikan survei dengan kebutuhan analisis lebih lanjut. Proyek *enhancement* ini juga merupakan hasil kerja sama antara peserta magang dengan salah satu rekan kerja di Tim CCF.

#### 1. Drop Source File



**Gambar 3.20** *Source File Survey Gallup di Server*

Masuk ke tahap *drop source file* pada Gambar 3.20, seluruh data Survey Gallup telah disimpan di server MobaXterm. *File* tersebut berbentuk format .csv dan .xlsx untuk setiap tipe pertanyaan yang ada seperti BE, BR, ITB, dan Problem. Rekan kerja membantu menyiapkan sebuah *file* baru yang dibutuhkan, yakni kategori IE (Interaction Experience) agar nantinya dapat digunakan sebagai *source* pada proses *mapping* di DWH. Pembuatan *file* IE dilakukan dengan menggunakan *script* Shell yang otomatis menghasilkan .csv dan .xlsx dengan format dan struktur data sesuai dengan *requirement*.

#### 2. Development



seperti nama dan wilayah cabang, beserta kodenya. Kemudian, kolom untuk tipe pertanyaan yang mana akan berisi nilai ‘Interaction Experience’ untuk menandakan data tersebut berasal dari survei IE. Terdapat pula kolom verbatim yang berisikan komentar-komentar atau *feedback* yang diberikan oleh *customer*. Tentunya, tabel dilengkapi tanggal *last update* sebagai penanda tanggal terakhir data masuk dan diperbarui dalam tabel.

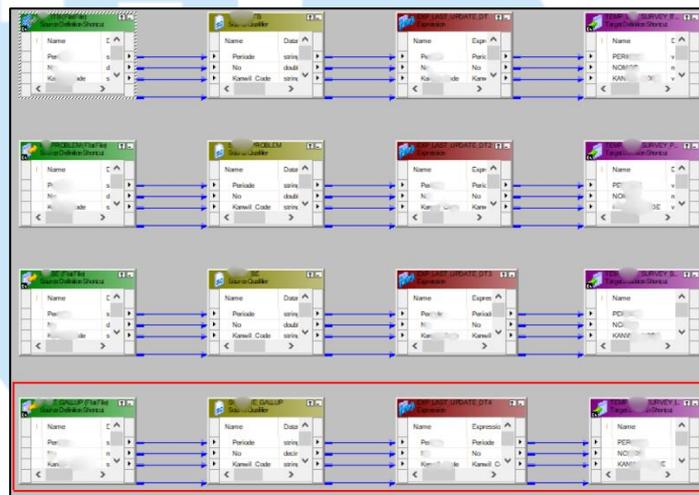
```
CREATE EXTERNAL TABLE raw.SURVEY_IE (
  PEF STRING,
  NC INT,
  YR_CODE STRING,
  K_CODE STRING,
  BRANCH_CODE STRING,
  BRANCH_NAME STRING,
  CITYTYPE STRING,
  Q7 DECIMAL(2,0),
  QUANTIFIER STRING,
  VER STRING,
  CI DECIMAL(11,0),
  LAST_UPDATE_TIMESTAMP)
PARTITIONED BY (INSERTED_DT STRING)
CLUSTERED BY (PEF) INTO 1 BUCKETS
STORED AS PARQUET
LOCATION '/user/hive/warehouse/raw/SURVEY_IE'
TBLPROPERTIES("parquet.compression"="SNAPPY");
```

**Gambar 3.22** Query DDL Create Table Raw Survey IE Big Data

Seperti biasa, pembuatan *table* juga harus dilakukan pada lingkungan Big Data untuk keperluan *offload* dan *backup*. Hal ini dilakukan dengan membuat tabel *raw* Survey IE, seperti pada Gambar 3.22. Struktur kolom pada tabel *raw* secara umum sama dengan tabel *temp* di Oracle, mencakup informasi kantor cabang, tipe pertanyaan IE, verbatim, dan tanggal pembaruan data. Namun, pada Hive terdapat penambahan kolom, yaitu *INSERTED\_DT* yang dipartisi sebagai tanggal data masuk ke dalam tabel Hive. Selain itu, tabel juga di-*bucket* berdasarkan kolom periode untuk mengoptimalkan proses *select query*. Pembuatan tabel dilakukan menggunakan *EXTERNAL TABLE* agar struktur tabel tidak akan terhapus meskipun datanya di-*delete*. Lokasi penyimpanan tabel ini diarahkan ke direktori *xxx/xxx/raw/SURVEY\_IE* pada *server* Big Data.

b. Mapping Informatica PowerCenter

Dalam melakukan *development mapping* Survey Gallup, dilakukan pembuatan dua *layer mapping* tabel Temp. *Mapping* tersebut sudah *existing* sebelumnya dengan terdapat *source* BR (Branch Relationship). Namun BR ingin dihapus di *mapping* dan digantikan dengan *source* IE.

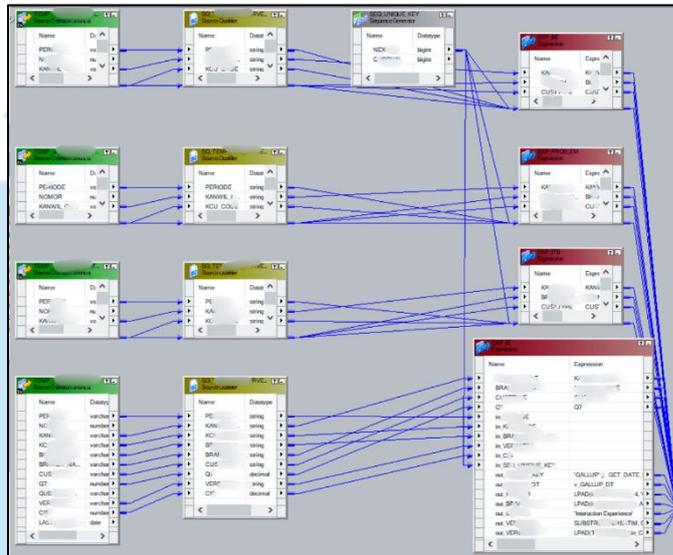


Gambar 3.23 Mapping Survey Gallup Temp 1

Gambar 3.23 menunjukkan *mapping* untuk Temp 1 Survey Gallup pada *tools* Informatica PowerCenter. Pada *mapping* ini, terlihat bahwa terdapat beberapa *source flatfile* Survey Gallup (berformat .csv) yang mempresentasikan masing-masing tipe pertanyaan, yaitu Branch Experience (BE), Intent To Buy (ITB), Problem, dan Interaction Experience (IE) yang menggantikan *source* BR. *Changes* yang dilakukan peserta magang pada *mapping* hanya menambah *source* IE dan mengalirkannya ke tabel *temp* Survey IE.

Setiap Source Qualifier hanya melewati satu tahapan transformasi Expression berupa penambahan kolom LAST\_UPDATE\_DT yang di *hardcode* menggunakan *function* SESSSTARTTIME. Setelah itu, langsung diarahkan ke tabel target *temporary* masing-masing. Dengan begitu, akan terbentuk beberapa tabel *temp* yang menampung data tipe

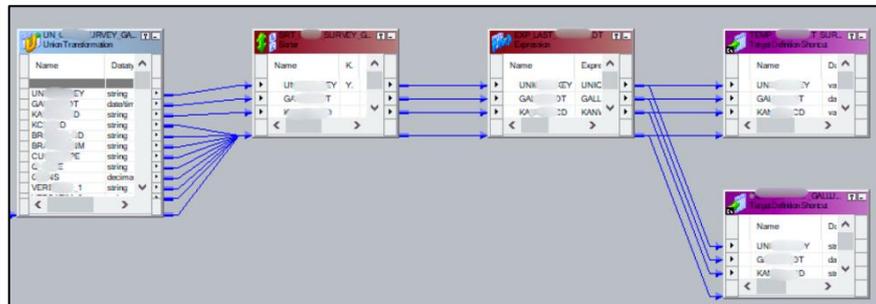
pertanyaan BE, Problem, ITB, dan IE. *Mapping temp* tersebut akan diatur dengan Pre SQL TRUNCATE TABLE setiap *mapping* dijalankan.



**Gambar 3.24** Mapping Survey Gallup Temp 2 (1)

Pada *mapping* Temp 2 (Gambar 3.24), seluruh tabel target hasil *mapping* Temp 1 dari masing-masing tipe survei akan dijadikan *source* data pada Temp 2. Seluruh Source Qualifier mula-mula akan dialirkan ke transformasi Expression terlebih dahulu, dimana terdapat berbagai *logic* yang diterapkan. Pertama, dibuat sebuah kolom bary untuk *unique key* yang dibentuk dengan cara menggabungkan String ‘GALLUP’, *function* GET\_DATE untuk mengambil tanggal, serta transformasi dari Sequence Generator untuk memberikan nomor urut unik, kemudian diformat dengan empat digit nol didepan nomor tersebut. Kedua, dilakukan pula pemformatan ulang pada kode cabang dan kode KCU dengan menggunakan *function* LPAD, yakni mengisi nol didepan angka agar total digitnya menjadi empat. Kolom tipe pertanyaan di-*harcode* menjadi String ‘Interaction Experience’ sebagai penanda survei baru. Terakhir, kolom *verbatim* diproses dengan *function*

SUBSTR hingga memiliki maksimal 1000 karakter dan dipecah menjadi sepuluh kolom *verbatim* baru.



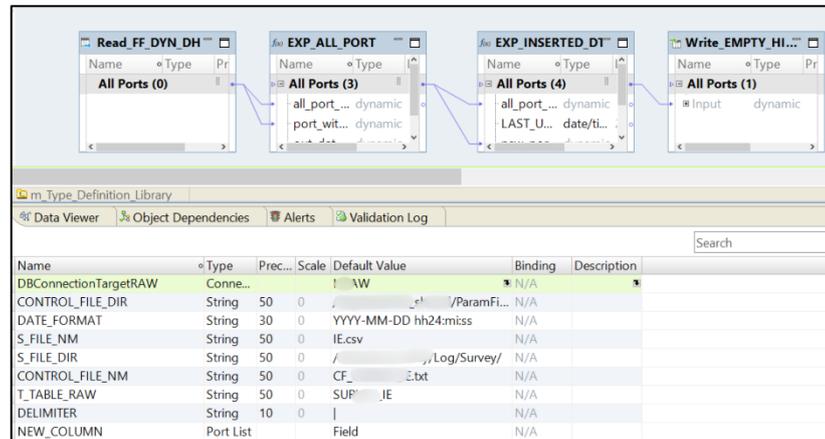
Gambar 3.25 Mapping Survey Gallup Temp 2 (2)

Setelah seluruh masing-masing tipe pertanyaan Survey Gallup diberikan *logic* pada Expression, maka selanjutnya mereka akan digabungkan menggunakan transformasi Union yang menandakan seluruh data dari berbagai tipe akan disatukan tanpa kondisi apa pun. Data hasil UNION ALL ini kemudian dialirkan ke transformasi Sorter untuk diurutkan berdasarkan *unique key*-nya secara Ascending. Ditambahkan transformasi Expression untuk mengisi kolom tanggal *last update* dengan SESSSTARTTIME. Proses *mapping* diakhiri dengan Target Definition, yaitu membuat tabel Temp Log Customer Survey Gallup sebagai tujuan akhir (target) dari hasil transformasi Temp 2.

### c. Mapping Informatica Developer

Pada *environment* Big Data yang menggunakan Informatica Developer, masing-masing tipe survei, seperti BR, ITB, BE, dan Problem sudah memiliki *mapping raw*-nya tersendiri untuk memuat data mentah berupa *flatfile* berformat .csv ke dalam sebuah tabel *raw* Big Data. Tipe survei BR yang sebelumnya sudah memiliki *mapping raw* serupa juga akan dihapus dan digantikan oleh *mapping raw* survei IE, khususnya pada *workflow* Survey Gallup di Informatica Developer. Namun, berbeda dari *mapping raw* biasanya, survei IE memanfaatkan

*dynamic mapping* sehingga *mapping* hanya cukup dijalankan berdasarkan parameter yang telah ditentukan dan tidak bersifat statis.



**Gambar 3.26** *Dynamic Mapping Survey IE Big Data*

*Dynamic mapping* yang digunakan pada Gambar 3.26 merupakan sebuah *template mapping* yang telah disiapkan pada sebuah *folder sharing* sehingga seluruh *developer* di DWH dapat mengakses dan menggunakannya. *Mapping* ini bersifat *reuseable* untuk berbagai kebutuhan *load* data ke tabel *raw* tanpa perlu mengatur ulang struktur *mapping*. *Dynamic mapping* hanya diwakili oleh satu *port* utama, yakni *all\_port* sebagai seluruh kolom dari *flatfile* yang akan diproses. Untuk keperluan pengujian, *mapping* ini digandakan dan ditempatkan pada *folder* pribadi peserta magang agar dapat diuji coba sebelum dijalankan lebih lanjut.

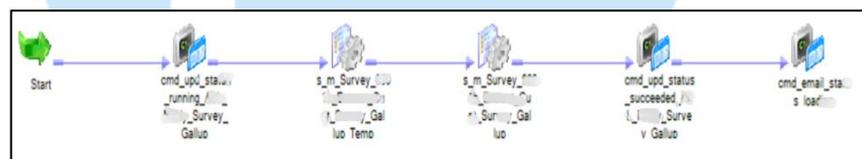
Terdapat beberapa parameter penting, seperti *DBConnectionTargetRAW* yang menunjukkan koneksi dari tabel target *raw*, *CONTROL\_FILE\_DIR* berisi *value* direktori penyimpanan *file* kontrol Survey IE di server, *DATE\_FORMAT* untuk mengatur format tanggal, *S\_FILE\_NM* dan *S\_FILE\_DIR* menunjukkan nama dan direktori *flatfile* yang akan di-*load*. Kemudian terdapat *CONTROL\_FILE\_NM* sebagai nama *file* kontrol Survey IE, *T\_TARGET\_RAW* berisi *value* tabel target *raw* Survey IE,

DELIMITER menentukan pemisah data dalam *flatfile*, dan NEW\_COLUMN bila ingin menambah kolom baru secara dinamis. Seluruh parameter ini akan ditembak dan diimplementasikan secara penuh di level *workflow*.

### 3. Testing Development

Proses *development* sudah selesai, tahap berikutnya ialah melakukan uji coba atau User Acceptance Testing untuk memastikan hasil *development* sesuai dengan kebutuhan. Skenario pengujian diberikan oleh Tim GPOL UAT sebagai acuan untuk melakukan validasi terhadap *development*.

#### a. Workflow Informatica PowerCenter



Gambar 3.27 Workflow Survey Gallup PowerCenter

Pada *workflow* di PowerCenter Workflow untuk *job* Survey Gallup, alur eksekusi diatur secara berurutan (tidak paralel), seperti pada Gambar 3.27. Hal ini membuat setiap proses dijalankan satu per satu sesuai urutan yang telah ditentukan dan saling bergantung satu sama lain. *Workflow* ini dimulai dari Start yang kemudian diarahkan ke sebuah Command untuk memberi perintah berupa *update* status menjadi '*running*' ketika *job* Survey Gallup mulai dieksekusi. Setelah itu, masuk ke Session *mapping* untuk Temp Survey Gallup, di mana *mapping* tersebut akan menghasilkan target *table* yang nantinya digunakan sebagai *source* pada proses berikutnya. Dari Session Temp, alur diteruskan ke Session *mapping* Survey Gallup keseluruhan yang menandakan bahwa Session ini memiliki *dependency* terhadap Session sebelumnya. Proses *workflow* dilanjutkan dengan Command untuk meng-*update* status eksekusi apabila *job* berjalan dengan baik dan

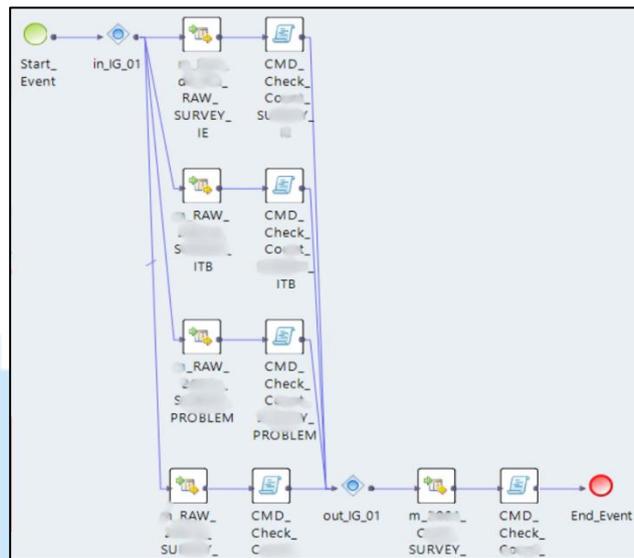
sukses. Terakhir diakhiri dengan Command untuk mengirim *email* notifikasi status *loading* kepada pihak terkait.

Workflow Run	Start Time	Completion Time	Status
wf_..._Survey_Gallup	5/10/2025 5:16:27 PM	5/10/2025 5:16:33 PM	Succeeded
s_m_Survey_G..._Fact_Surve...	5/10/2025 5:16:27 PM	5/10/2025 5:16:33 PM	Succeeded
wf_..._Survey_Gallup	5/10/2025 5:16:17 PM	5/10/2025 5:16:25 PM	Succeeded
s_m_Survey_G..._Fact_Surve...	5/10/2025 5:16:17 PM	5/10/2025 5:16:25 PM	Succeeded

**Gambar 3.28** Status Eksekusi Session Survey Gallup

Kedua Session Survey Gallup yang telah dimasukkan pada *workflow* akan dijalankan dan dieksekusi untuk menghasilkan *table datamart*. Hasil eksekusi dari masing-masing *session* akan dimonitor melalui aplikasi PowerCenter Monitoring, seperti pada Gambar 3.28 *Tools* ini akan menampilkan status akhir eksekusi berupa Succeeded atau Failed, serta menyediakan *log* detail yang digunakan untuk memeriksa proses apabila terjadi *error*. Seluruh *session* terkait proyek penalti (*temp 1*, *temp 2*, dan *fact*) telah berhasil dijalankan dengan status Succeeded tanpa adanya *error*. Hal ini menunjukkan proses ETL telah berjalan sesuai *flow* yang dirancang, serta data telah berhasil dimuat ke masing-masing tabel target, khususnya tabel akhir Survey Gallup.

b. Workflow Informatica Developer



**Gambar 3.29** Workflow Survey Gallup Informatica Developer

Gambar 3.29 menunjukkan *workflow* pada Informatica Developer untuk mengatur seluruh alur eksekusi *mapping* Survey Gallup. Setiap *mapping* tabel Raw BE, ITB, Problem, dan IE akan dimasukkan pada satu alur *workflow* yang sama. *Mapping* BR yang sebelumnya terdapat pada *workflow* ini, digantikan oleh *dynamic mapping* untuk Survey IE. Alur dimulai dari Start Event, kemudian dilanjutkan ke Inclusive Gateway sebagai percabangan ke masing-masing *mapping* Raw. Setiap *mapping* Raw tersebut dipasangkan dengan Command *check count* untuk memastikan validitas jumlah data. Setelah semua *mapping* Raw dijalankan, alur kembali digabung melalui Inclusive Gateway kedua yang dilanjutkan ke *mapping* Survey Gallup keseluruhan.

Input	Value
Mapping Parameter Inputs	
SHAREFOLDER	
m_...	
DBConnectionTargetRAW	DBConnectionTargetRAW (Parameter Value)
CONTROL_FILE_DIR	CONTROL_FILE_DIR (Parameter Value)
DATE_FORMAT	DATE_FORMAT (Parameter Value)
S_FILE_NM	S_FILE_NM (Parameter Value)
S_FILE_DIR	S_FILE_DIR (Parameter Value)
CONTROL_FILE_NM	CONTROL_FILE_NM (Parameter Value)
T_TABLE_RAW	T_TABLE_RAW (Parameter Value)
DELIMITER	DELIMITER (Parameter Value)
NEW_COLUMN	
Mapping Task Configuration Properties	
Mapping Task Log Directory	LogDir (Parameter Value)

**Gambar 3.30** Parameter pada Workflow Survey Gallup

Pada level *workflow* ini, seluruh parameter yang telah diuji coba sebelumnya di level *mapping* akan didefinisikan, seperti pada Gambar 3.30. Bagian bawah *canvas* terdapat pengaturan untuk parameter-parameter yang ingin dimasukkan. Di sinilah seluruh *value* parameter akan diatur persis seperti yang dilakukan pada *dynamic mapping* sebelumnya, mulai dari koneksi *database*, direktori *file*, target *table*, kontrol *file*, *delimiter*, dan lain sebagainya.

a. Hasil Datamart

**Gambar 3.31** Hasil Datamart Enhancement Survey Gallup

Gambar 3.31 menunjukkan hasil proses data Survey Gallup yang telah disesuaikan dan di-*enhance*. Sebelah kiri menampilkan data yang berhasil masuk ke tabel *temp* di Oracle sedangkan sebelah kanan menunjukkan hasil yang sama di Big Data. Data dari berbagai tipe survei, termasuk Interaction Experience telah berhasil diproses sesuai

dengan *requirement user*. Nantinya, *datamart* ini akan digunakan oleh Tim SPECTA. Selama fase User Acceptance Testing, hasil pemrosesan data secara rutin disampaikan kepada Tim UAT untuk dilakukan verifikasi. Setelah itu, proyek ini resmi diimplementasi ke *environment Production* pada tanggal 22 April 2025.

### 3.2.3 Proyek Penurunan Data Log Utilisasi OCR Engine Glair

Optical Character Recognition (OCR) merupakan sebuah teknologi yang digunakan untuk mengekstraksi berbagai teks dari dokumen digital, seperti *file* PDF atau gambar sehingga teks tersebut dapat dibaca dan diproses oleh sistem secara otomatis. Teknologi ini mampu menangani dokumen-dokumen yang masuk dari berbagai *channel*. Proyek penurunan *log* menjadi bagian dari *monitoring* performa teknologi OCR. Dalam proyek ini, Data Management C berperan sebagai Product Owner karena memiliki tanggung jawab langsung terhadap kelengkapan, keakuratan, dan kebermanfaatan data *log* dari OCR Engine.

Proyek hadir dikarenakan terdapat penambahan *engine* baru, yaitu Engine Glair sebagai salah satu varian OCR Engine yang digunakan. Sebelumnya, OCR sudah memiliki *engine-engine*, seperti Abby Sync, Static Grha, Engine DTM, dan lain-lain. Proyek ini merupakan proyek *existing*, di mana telah memiliki *pipeline* penurunan *log* untuk *engine* OCR yang lainnya. Namun, perlu dilakukan penyesuaian agar *log* utilisasi dari Engine Glair juga dapat diturunkan dan di-*monitor* melalui Data Warehouse (DWH). Penurunan *log* OCR Engine Glair juga melibatkan kerja sama dengan Tim Data Standardization di DWH untuk mengonversi dan menyiapkan *source* data dalam bentuk *file* AVRO selama proses *development*.

1. Source File

id_from_channel	channel	filename	pages	start_time	end_time	ocr_duration	status	reason
...	...	...	...	...	...	...	...	...

Gambar 3.32 Source Table Temp AVRO OCR

Penurunan *source* data selama proses *development* dilakukan oleh Tim Data Management ke Tim Data Standarization di DWH. Sebelum data digunakan untuk pengembangan lebih lanjut, Tim Data Standardization terlebih dahulu memproses data mentah tersebut untuk dikonversi ke dalam format *file* AVRO. AVRO merupakan sebuah format *file* berbasis Schema yang menggunakan *row-based storage* sehingga cocok untuk menyimpan data dalam *environment* Big Data, khususnya pada Hadoop. Setelah dikonversi, *file* AVRO tersebut kemudian dimuat ke dalam tabel Temp Avro Userfile di Big Data, seperti pada Gambar 3.32. Tabel inilah yang nantinya akan diakses dan diimpor oleh peserta magang untuk menjadi *source table* dalam *development mapping* log utilisasi OCR Engine Glair.

## 2. Development

Proses *development* disini secara garis besar ialah menambahkan *source file* AVRO tersebut serta menambah *logic-logic* khusus untuk *engine* terbaru, yaitu Engine Glair ke dalam *mapping* Log OCR Engine yang telah ada sebelumnya. Seperti biasanya, pengembangan *mapping* dilakukan pada *environment* Oracle dan Big Data. *Development* tidak memerlukan pembuatan *table* baru, melainkan hanya menambah kolom Accuracy pada *table* Log OCR Engine di Big Data.

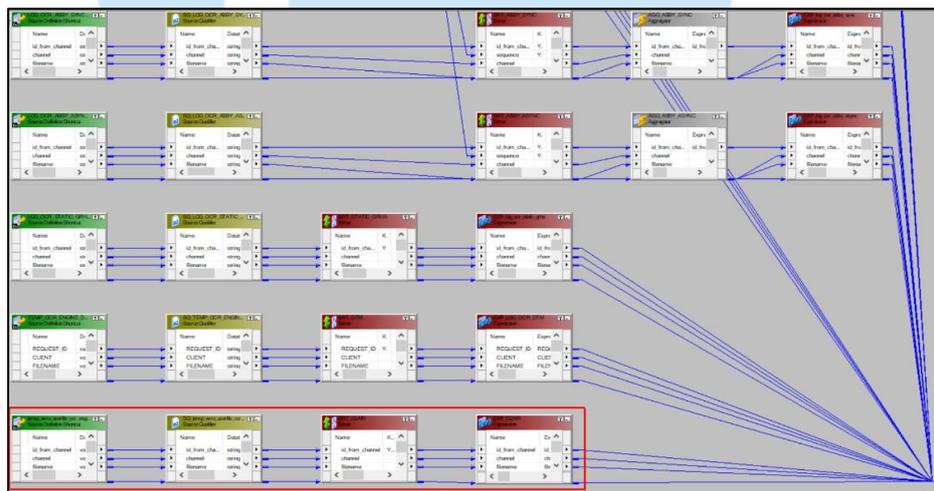
### a. Penambahan Kolom pada Table Big Data

```
ALTER TABLE bcadw.OCR_ENGINE_LOG_FACT ADD COLUMNS (ACCURACY DECIMAL(38,14));
ALTER TABLE bcadw.TF_OCR_ENGINE_LOG_FACT ADD COLUMNS (ACCURACY DECIMAL(38,14));
```

Gambar 3.33 Penambahan Kolom Accuracy Big Data

Proses *development* pertama kali akan dilakukan penambahan kolom baru, yaitu ACCURACY pada struktur tabel *fact* OCR di Big Data, baik itu di table *source* maupun target. Hal ini dikarenakan tabel tersebut pada Big Data belum memiliki informasi mengenai tingkat keakuratan hasil pembacaan OCR Engine. Penambahan kolom menggunakan *query* ALTER TABLE dan kolom ACCURACY didefinisikan dengan tipe data Decimal (38,14) yang berarti kolom tersebut menyimpan angka desimal dengan total 38 digit, termasuk 14 digit di belakang koma, seperti yang dapat dilihat pada Gambar 3.33.

a. Mapping Informatica PowerCenter



Gambar 3.34 Mapping OCR Engine Log Fact Temp 1 (1)

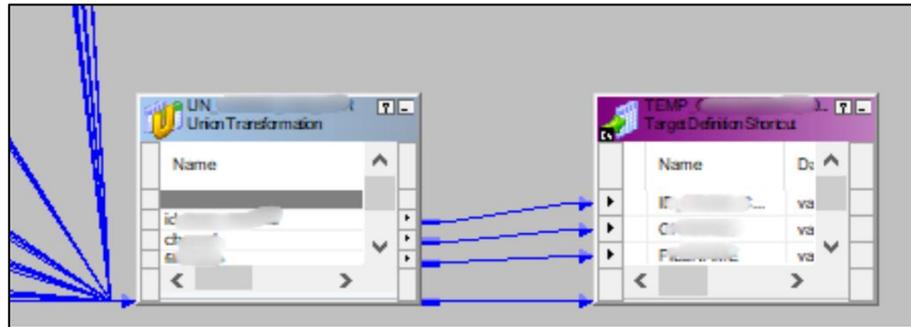
Pada *mapping* OCR Engine Temp 1, langkah pertama yang dilakukan adalah mengimpor *source* AVRO yang berisi *log* dari Engine Glair ke *mapping* Log OCR *existing* di PowerCenter Designer. Meskipun PowerCenter Designer umumnya terintegrasi langsung dengan *environment* Oracle, tetapi PowerCenter Designer juga tetap mendukung penggunaan *table* dari *environment* Big Data sebagai *source*. Seperti yang dilihat pada Gambar 3.34, *mapping* ini sudah memiliki berbagai pemetaan untuk *engine* OCR lain yang telah dilakukan sebelumnya. Setelah *source table* AVRO dimasukkan ke

Source Qualifier, data akan dialirkan terlebih dahulu ke transformasi Sorter untuk mengurutkan data secara *ascending* berdasarkan kolom ID\_FROM\_CHANNEL. Kemudian, data dimasukkan ke dalam transformasi Expression yang digunakan untuk menambah berbagai *logic* serta mengubah tipe data dari beberapa kolom. Hal ini diperlukan karena *source* AVRO yang di-*convert* ke *table* umumnya terbaca sebagai String pada seluruh kolomnya sehingga perlu penyesuaian tipe data dan panjang (*length*) kolom sesuai dengan *requirement*.

15	in_pages	decimal	15	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TO_DECIMAL(pages, 0)
16	out_start_time	date/time	29	9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TO_DATE(in_start_time, 'DD-MM-YYYY HH24.MI.SS')
17	in_end_time	date/time	29	9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TO_DATE(in_end_time, 'DD-MM-YYYY HH24.MI.SS')
18	in_receive_request	decimal	38	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TO_DECIMAL(in_receive_request, 2)
19	out_receive_request	decimal	38	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TO_DECIMAL(in_receive_request, 2)
20	out_start_time	date/time	29	9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TO_DATE(out_start_time, 'DD-MM-YYYY HH24.MI.SS')
21	out_end_time	date/time	29	9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TO_DATE(in_end_time, 'DD-MM-YYYY HH24.MI.SS')

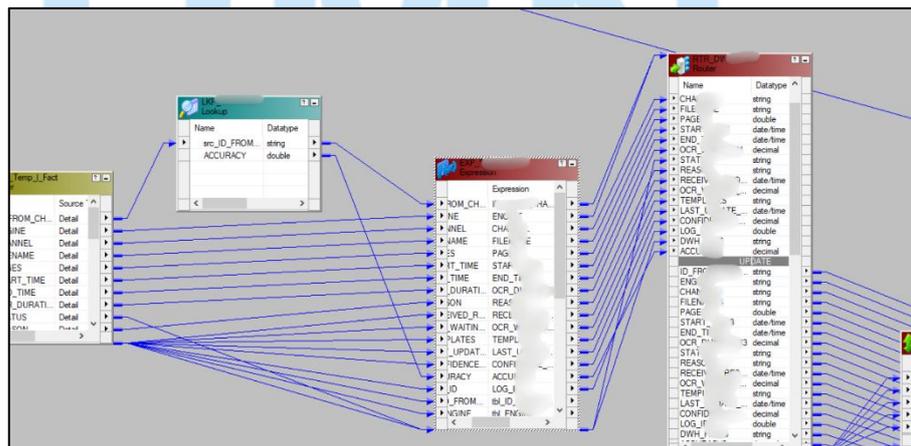
**Gambar 3.35** Expression Perubahan Tipe Data

Gambar 3.35 menunjukkan transformasi Expression yang dilakukan untuk mengubah semua tipe data bernilai String akibat format AVRO. Beberapa kolom seperti START\_TIME, END\_TIME, dan RECEIVE\_REQUEST seharusnya memiliki tipe data tanggal sehingga harus diubah ke format DATE dengan pola DD-MM-YYYY HH24:MI:SS (format yang dapat dibaca oleh Informatica). Kolom numerik lainnya akan diubah ke tipe data Decimal dengan *length* sebesar 38 dan *precision* dua angka di belakang koma. Untuk kolom PAGES diubah ke Decimal tanpa adanya angka di belakang koma (0 digit desimal). Selain itu, setiap kolom hasil konversi ini selalu diberikan *prefix* 'out\_' pada nama *port*-nya untuk menandai bahwa *port* tersebut merupakan hasil yang akan digunakan sebagai *output* ke transformasi selanjutnya. Sementara kolom asli bertipe String akan tetap disertakan dalam Expression sebagai *input-only* (referensi *value*) tanpa mencentang *checkbox output*.



Gambar 3.36 Mapping OCR Engine Log Fact Temp 1 (2)

Terlihat pada Gambar 3.36, setelah melalui transformasi Expression dan seluruh tipe data berhasil disesuaikan, data dari Engine Glair akan dialirkan ke transformasi Union. Disini, terdapat penambahan satu grup baru khusus untuk Engine Glair agar dapat digabungkan dengan data dari *engine* lainnya yang sudah terlebih dahulu ada pada *mapping*. Transformasi Union berfungsi untuk menggabungkan seluruh data dari berbagai *engine* ke dalam satu aliran data tanpa terkecuali. Setelah itu, data gabungan tersebut akan diarahkan menuju Target Definition sebagai hasil akhir tabel *layer* Temp 1.



Gambar 3.37 Mapping OCR Engine Log Fact Temp 2

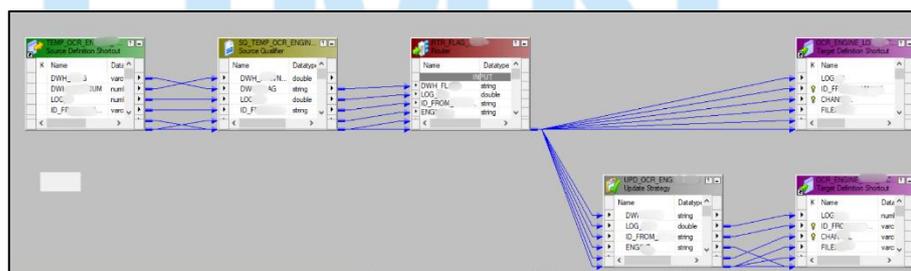
Pada *mapping* OCR Engine Log Fact Temp 2, peserta magang hanya melakukan *changes* untuk *logic* transformasi Expression,

tepatnya di *port* STATUS. Logika tersebut dimodifikasi agar Engine Glair dapat ikut diproses menggunakan aturan-aturan yang sudah berlaku untuk Engine lainnya. Berikut *logic* yang berlaku pada *port* STATUS.

```
DECODE(TRUE, IN(ENGINE, 'engine_dtm', 'engine_glair') AND in_STATUS='DPA-000', 'Success', IN(ENGINE, 'engine_dtm', 'engine_glair') AND in_STATUS != 'DPA-000', 'Failed', in_STATUS)
```

**Gambar 3.38** *Logic STATUS OCR Engine Log Fact*

Gambar 3.38 menunjukkan logika menggunakan *function* DECODE yang fungsinya mirip seperti IF-ELSE dengan kondisi Boolean. Pertama, jika nilai kolom ENGINE termasuk ke dalam 'engine\_dtm' atau 'engine\_glair' dan STATUS bernilai 'DPA-000', maka *output* STATUS akan berubah menjadi 'Success'. Sebaliknya, jika ENGINE termasuk 'engine\_dtm' atau 'engine\_glair' tetapi STATUS tidak sama dengan 'DPA-000', maka nilai *output* STATUS akan tetap berubah menjadi 'Failed'. Hal ini menandakan bahwa kode 'DPA-000' merupakan indikator yang penting untuk menentukan keberhasilan proses dari sebuah OCR Engine.

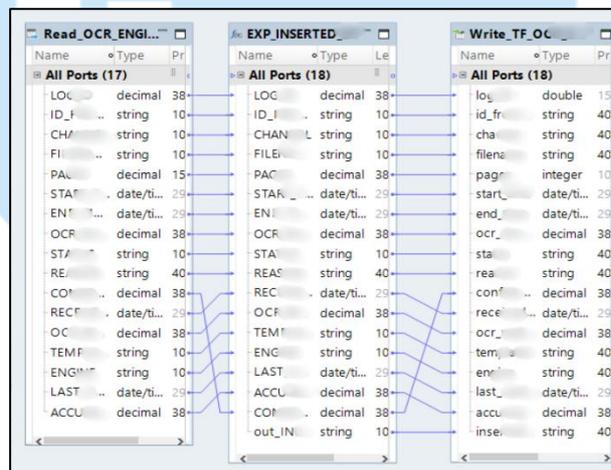


**Gambar 3.39** *Mapping OCR Engine Log Fact*

Pada *mapping* OCR Engine Fact (*layer* terakhir), target *table* pada Temp 2 akan dijadikan *source*. Secara garis besar, alur *mapping* ini akan dimulai dengan transformasi Router yang berfungsi untuk memisahkan data berdasarkan nilai *flag* yang sebelumnya telah ditentukan pada *temp* 2, yaitu *flag* 'U' untuk *update* dan 'N' untuk *new*.

Data dengan *flag* 'U' akan dialirkan ke transformasi Update Strategy yang mengatur data untuk melakukan aksi pembaruan pada *record* yang sudah ada di tabel *fact*. Sementara itu, data dengan *flag* 'N' langsung diteruskan ke *target table* untuk penambahan data baru. Walaupun terdapat dua jalur yang data dan dua transformasi Target Definition, keduanya tetap akan bermuara pada tabel *fact* yang sama, yaitu OCR Engine Log Fact sehingga perubahan serta penambahan data tetap dalam satu kesatuan tabel yang sama.

b. Mapping Informatica Developer



Gambar 3.40 Mapping OCR Engine Log Fact Big Data

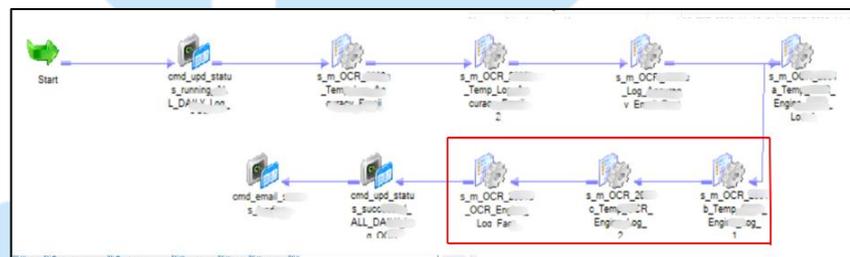
Gambar 3.40 menunjukkan proses *offload* data dari tabel *fact* OCR Engine Oracle ke tabel *fact* Big Data di Informatica Developer. Meskipun sebelumnya kolom baru ACCURACY sudah ditambahkan melalui perintah ALTER TABLE, kolom tersebut tetap perlu ditambahkan secara manual di Informatica agar bisa dikenali dalam proses *mapping*. Penambahan kolom tersebut dilakukan pada *source* Read (*fact* Oracle) dan target *offload* Write (*fact* Big Data). Setelah kolom ditambahkan secara manual, barulah struktur kolom antara *source* dan target dapat selaras dan tersinkronisasi. *Mapping* juga

dilakukan transformasi Expression untuk memberi *logic* pada *inserted date*.

### 3. Testing Development

Tahap selanjutnya ialah melakukan *testing* atau UAT terhadap *mapping* OCR Engine Log Fact yang telah selesai dikembangkan. Selama masa UAT, beberapa pihak dari tim lain juga turut melakukan pengujian secara paralel, salah satunya dari Tim Data Standardization terkait penyediaan dan pemrosesan data turunan dalam bentuk AVRO (*datalake*) yang kemudian dapat dilanjutkan proses *testing* oleh peserta magang pada sisi Tim CCF untuk menghasilkan *datamart*. Skenario pengujian diberikan oleh Tim GPOL UAT sebagai acuan untuk melakukan verifikasi terhadap *development*.

#### a. Workflow Informatica PowerCenter



**Gambar 3.41** Workflow OCR Engine Informatica PowerCenter

Gambar 3.41 menampilkan keseluruhan alur *job* dari OCR Engine yang tergabung dalam satu *workflow* utama. *Workflow* tersebut memuat berbagai *session* dari masing-masing proyek OCR yang saling berurutan dan saling bergantung satu sama lain. Namun, tidak terdapat perubahan apapun pada *session-session* selain ketiga *session* terkait OCR Engine Log yang mengalami perubahan karena penambahan Engine baru, seperti yang ditunjukkan pada gambar dengan kotak berwarna merah. Salah satu perubahan yang penting dilakukan, ialah mengatur koneksi di *session* Temp 1, khususnya pada bagian *source*. Hal ini dikarenakan *source* dari *session* Temp 1

merupakan *table* Temp AVRO yang berasal dari *environment* Big Data sehingga koneksi *default* akan menyebabkan *error* jika tidak diatur. Maka dari itu, *connection* pada *source* AVRO harus diubah menjadi Hive ODBC agar dapat membaca data dari *table* Big Data di Informatica PowerCenter. Pada *environment* Big Data, tidak ada *workflow* yang di-*development* oleh peserta magang sehingga pemrosesan data hanya dilakukan melalui *running mapping*.

Workflow Run	Start Time	Completion Time	Status
wf_..._OCR	5/25/2025 5:20:13 PM	5/25/2025 5:20:22 PM	Succeeded
s_m_OCR_2..._ine_Log_F...	5/25/2025 5:20:13 PM	5/25/2025 5:20:20 PM	Succeeded
wf_..._OCR	5/25/2025 5:18:20 PM	5/25/2025 5:20:08 PM	Succeeded
s_m_OCR_...mp_OCR_Engine...	5/25/2025 5:18:20 PM	5/25/2025 5:20:08 PM	Succeeded
wf_E..._OCR	5/25/2025 5:18:07 PM	5/25/2025 5:18:17 PM	Succeeded
s_m_OCR_...mp_OCR_Engine...	5/25/2025 5:18:07 PM	5/25/2025 5:18:16 PM	Succeeded

**Gambar 3.42** Status Eksekusi Job OCR Engine Log Fact

Gambar 3.42 menunjukkan status eksekusi dari *session-session* untuk OCR Engine Log Fact di PowerCenter Monitor. Meskipun dalam *workflow* tersebut terdapat banyak *session*, tetapi hanya *session* yang berkaitan langsung dengan pembentukan *fact table* OCR Engine Log yang dieksekusi. Ketiga *session* tersebut berupa *session* Temp 1, Temp 2, dan Fact yang secara berurutan memroses data dari *source* AVRO. Seluruh *session* tersebut telah berhasil dijalankan dengan status Succeeded tanpa adanya *error*. Hal ini dapat dilanjutkan dengan memverifikasi hasil Datamart *fact table* apakah sudah sesuai dengan kebutuhan *user*.

b. Hasil Datamart



1) dengan dua hari yang lalu (saldo H-2). Apabila hasilnya negatif, maka itu menandakan bahwa terjadi penurunan saldo oleh nasabah-nasabah tertentu.

Hasil *datamart* akan mengidentifikasi 10 nasabah teratas (*top 10*) yang mengalami penurunan CASA terbesar pada Kantor Cabang Utama (KCU atau Kantor Cabang Pembantu (KCP) BCA tertentu. Selain itu, proyek ini juga dilakukan perhitungan Growth Ratio, yaitu persentase perubahan saldo dengan rumus berikut.

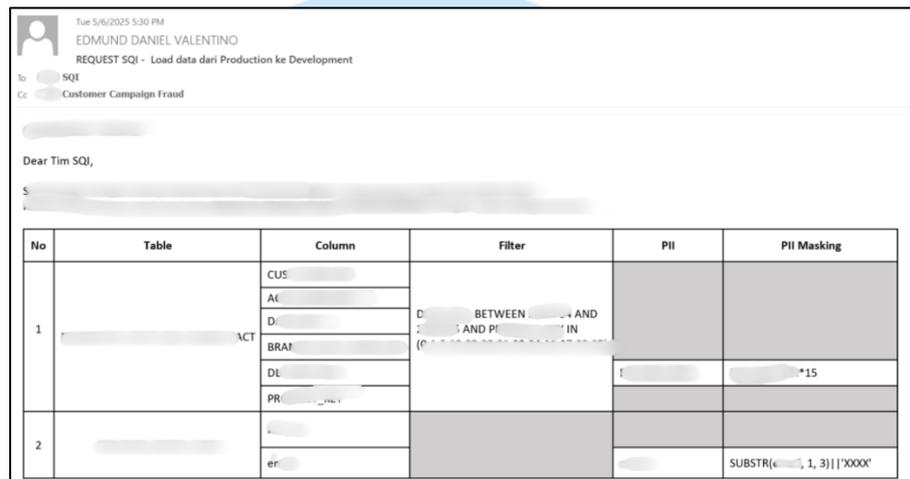
$$\text{Growth Ratio} = \frac{(\text{Saldo kemarin} - \text{Saldo dua hari lalu})}{\text{Saldo dua hari lalu}} \times 100\%$$

Oleh karena itu, *output* yang dihasilkan akan meliputi nama nasabah 10 teratas, nilai persentase Growth Ratio, dan informasi lokasi kantor cabang tempat nasabah membuka rekening. Sebenarnya, proyek ini merupakan proyek *existing* yang sudah pernah diimplementasikan sebelumnya. Namun, pada saat itu terdapat sebuah temuan saat implementasi di lingkungan *production*, yaitu nasabah yang baru membuka rekening pada H-1. Pada kasus tersebut, CUST CURR KEY H-1 akan tetap muncul, tetapi CUST CURR KEY H-2 akan bernilai Null karena nasabah baru membuka rekening kemarin sehingga menyebabkan kesalahan logika pada perhitungan total penurunan CASA maupun Growth Ratio. Maka dari itu, dilakukan *enhancement* sebagai solusi dengan mengisi CUST CURR KEY H-2 sama dengan CUST CURR KEY H-1 jika data CUST CURR KEY H-2 tidak tersedia agar tidak terjadi data *loss*.

#### 1. Request Penurunan Data Production ke Development

Pada proses *development mapping* Table Temp CASA, terdapat empat table yang akan digunakan sebagai *source*. Dari keempat *table* tersebut, dua diantaranya memerlukan proses penurunan/*load* data dari *environment* Production ke Development. Oleh karena itu, peserta magang melakukan permintaan *load* data kepada Tim Software Quality Improvement (SQI) supaya data yang digunakan dalam proses *development* akan dibuat semirip mungkin dengan kondisi aktual pada Production. Dua *table* yang di-*request*

adalah Account Balance Summary Fact, yaitu *fact table* yang berisi saldo harian nasabah di *database* Oracle dan Push Notif Teams berisi kode-kode cabang di Big Data.

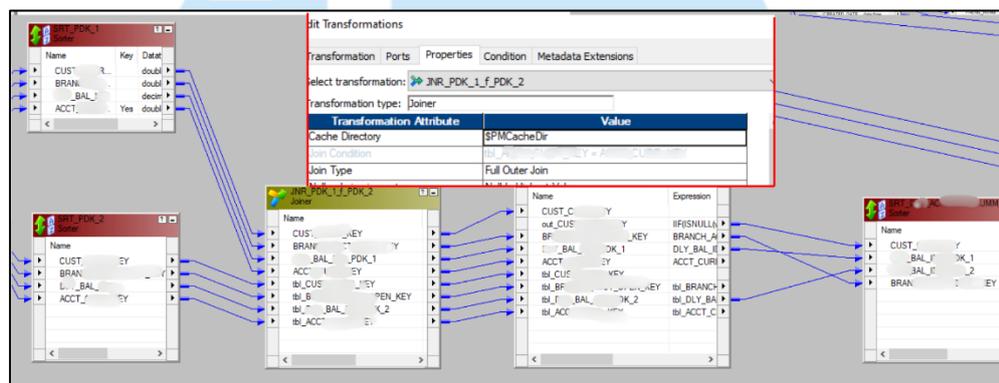


Gambar 3.44 Request Penurunan Data Production

Dalam melakukan *request* penurunan data ini, peserta magang harus terlebih dahulu mengisi formulir Microsoft Forms yang telah disediakan oleh Tim SQI dengan mencantumkan tujuan yang jelas terkait kebutuhan *load* data tersebut. Setelah itu, permintaan dilanjutkan dengan *follow-up* melalui email kepada Tim SQI, seperti pada Gambar 3.44 untuk menyampaikan informasi detail mengenai nama tabel, kriteria filter data, dan metode *masking* terhadap data pribadi atau kolom yang mengandung PII (Personally Identifiable Information). Filter yang digunakan ialah berdasarkan POSTING DATE KEY dan PRODUCT KEY tertentu agar data yang diturunkan sesuai dengan kasus proyek dan tidak menurunkan terlalu banyak jumlahnya. Karena data Production sangat sensitif, proses *masking* pun diterapkan untuk kolom saldo pada *table* Account Balance Summary Fact dengan mengalikan masing-masing nilainya dengan 15. Sementara itu, kolom EMAIL di *table* PUSH NOTIF TEAMS di-*masking* dengan menambahkan String 'XXX' pada masing-masing nilai. Hal ini membuat data yang diturunkan ke *environment* Development tetap aman dan sesuai dengan tujuan proyek.

## 2. Development

Perubahan yang dilakukan oleh peserta magang selama tahap *development* ialah membenarkan beberapa logika dalam *mapping* CASA di Informatica PowerCenter. *Development* hanya dilakukan pada *environment* Oracle karena hasil *datamart* terletak di *database* Oracle tanpa memerlukan *development* di Big Data.



Gambar 3.45 Mapping Push Notif CASA

*Mapping* pada Gambar 3.45, berawal dari *table* Account Balance Summary Fact yang dipecah datanya melalui transformasi Router untuk membagi menjadi dua kelompok, yaitu data saldo H-1 dan H-2 dengan key ACCT CURR KEY. Namun, perhitungan yang tepat terhadap saldo nasabah juga bergantung pada CUST CURR KEY-nya. Permasalahan muncul ketika terdapat nasabah baru yang hanya memiliki informasi saldo H-1 karena baru membuka rekening di hari tersebut sedangkan pada H-2, belum ada data terkait terutama CUST CURR KEY nasabah sehingga nilainya akan Null. Maka dari itu, solusinya ialah mengganti tipe Joiner yang sebelumnya Inner Join diubah menjadi Full Outer Join supaya seluruh kombinasi data H-1 dan H-2 tetap terjaga meskipun salah satu sisi tidak memiliki pasangan data.

```
IF(ISNULL(v_CUST_CURR_KEY), tBI_CUST_CURR_KEY, v_CUST_CURR_KEY)
```

Gambar 3.46 Logic Pengisian Nilai Null



Gambar 3.47 menunjukkan tabel skenario yang diberikan oleh *user* berisi beberapa skenario pengujian yang mencakup aspek penting dari *datamart* yang dihasilkan. Beberapa skenario tersebut antara lain memastikan bahwa informasi KCU dan KCP BCA tempat nasabah mengalami penurunan CASA sudah sesuai, validasi terhadap *email* yang berisi *null*, validasi nama nasabah, pengecekan terhadap *template code* berisi nilai yang sama untuk seluruh baris karena dilakukan *hardcode* pada *mapping*. Kemudian, dilakukan verifikasi terhadap jumlah nasabah yang turun, di mana maksimal jumlahnya ialah 10 dan apabila data penurunan kurang dari 10 nasabah, maka hasilnya akan menyesuaikan secara dinamis. Perhitungan terhadap penurunan CASA dan Growth Ratio juga dilakukan pengujian untuk memastikan rumus yang digunakan sudah benar. Seluruh hasil pengujian dibuktikan melalui tangkapan layar sebagai gambar pendukung dan keterangan tambahan dari masing-masing skenario.

```

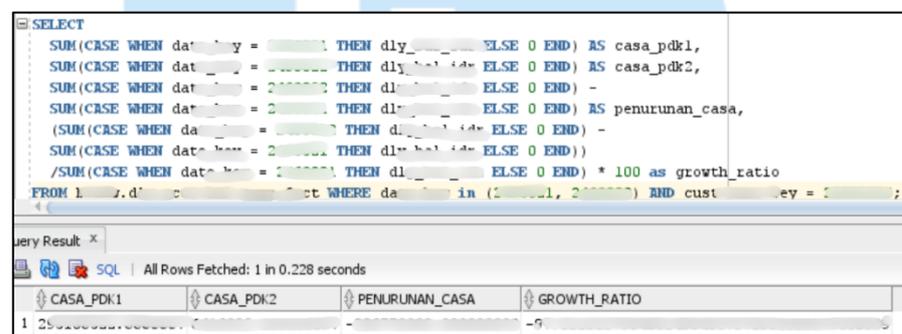
SELECT cu_..._key
FROM ..._dim_..._fact
WHERE cu_..._key
IN (SELECT cu_..._key
FROM ..._dim_..._dimension WHERE cu_..._name = '...');

```

**Gambar 3.48** Query Filter Cust Curr Key

Untuk melakukan verifikasi terhadap perhitungan Growth Ratio dan penurunan CASA pada hasil *datamart*, peserta magang perlu terlebih dahulu mengetahui CUST CURR KEY dari masing-masing nama nasabah yang muncul pada *datamart*. Hal ini penting karena informasi saldo nasabah pada H-1 dan H-2 hanya dapat ditelusuri melalui CUST CURR KEY yang tersimpan pada *table* Account Balance Summary Fact. Mula-mula, peserta magang mencari tahu CUST CURR KEY yang dimiliki nama nasabah tertentu melalui *table* Customer Currency Dimension. Setelah ditemukan, dilakukan pengecekan ke *table* Account Balance Summary Fact untuk melihat

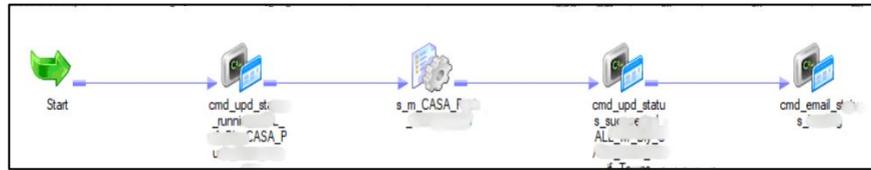
saldo tersebut. Namun, terdapat beberapa kasus, yaitu satu nama nasabah dapat memiliki sangat banyak CUST CURR KEY yang jumlahnya tidak memungkinkan untuk dicek satu per satu. Bahkan, hanya terdapat satu CUST CURR KEY yang benar-benar muncul di *fact table*. Maka dari itu, peserta magang menggunakan *query SQL*, seperti pada Gambar 3.48 untuk memfilter dan menemukan satu dari banyaknya CUST CURR KEY tersebut untuk satu nama nasabah pada tabel *fact* dengan cepat.



Gambar 3.49 Query Pengecekan Perhitungan CASA

Pada Gambar 3.49 peserta magang menggunakan *query SQL* tersebut untuk memvalidasi hasil perhitungan penurunan CASA dan Growth Ratio dari setiap nasabah yang muncul di hasil *datamart* menggunakan CUST CURR KEY-nya. Pertama, total saldo H-1 dan H-2 dijumlahkan secara terpisah dengan menggunakan agregat SUM yang dikondisikan berdasarkan DATE KEY. Setelah itu, penurunan CASA dihitung dengan cara mengurangi total saldo H-2 dengan H-1. Untuk memperoleh nilai Growth Ratio, selisih antara saldo H-2 dengan H-1 akan dibagi dengan saldo H-2 dan dikalikan 100 persen untuk mendapatkan persentase sehingga *query* tersebut akan menghasilkan data hasil perhitungan CASA.

b. Workflow Informatica PowerCenter



**Gambar 3.50** *Workflow Push Notif CASA Informatica PowerCenter*

*Workflow* untuk proyek penurunan CASA pada Gambar 3.50, diterapkan aturan baku berupa penggunaan Command Task untuk memperbarui status *workflow* ketika sedang dijalankan. *Workflow* hanya terdiri dari satu *session* atau *mapping* yang mana bertugas untuk menghasilkan *datamart* ke *table* Push Notif CASA tanpa adanya *layer-layer* transformasi tambahan seperti pada proyek lainnya. Hal tersebut menandakan *session* berdiri sendiri dan tidak memiliki ketergantungan dengan *session* manapun. Kemudian, alur dilanjutkan dengan Command Task untuk memperbarui status menjadi ‘*succeeded*’ apabila *session* berhasil dieksekusi tanpa *error*. Terakhir, terdapat Command Task untuk mengirim *email* berupa status bahwa data sedang diproses.

c. Hasil Datamart

EMAIL	TEMPLATE_CD	BRANCH_NM	GROWTH_RATIO
			sebesar 0%

**Gambar 3.51** *Datamart Push Notif CASA*

Gambar 3.51 menunjukkan hasil akhir *datamart* proyek data penurunan CASA nasabah yang terdapat informasi sesuai dengan kebutuhan *user*. Hal ini mencakup *email* Kantor Cabang Utama (KCU) atau Kantor Cabang Pembantu (KCP), beserta nama KCU/KCP nasabah yang mengalami penurunan tersebut. Kemudian, *datamart* juga memuat data berupa persentase Growth Ratio dari beberapa nama nasabah yang muncul (10 nasabah dengan penurunan CASA terbesar).

Persentase Growth Ratio pada kolom GROWTH RATIO ditampilkan dalam bentuk kalimat yang disusun secara dinamis, misalnya “Nasabah A sebesar 20%”, “Nasabah B sebesar 30%”, dan seterusnya. Kalimat tersebut dipisahkan dengan tag ‘<br>’ agar dapat ditampilkan dengan format baris terpisah saat dikonsumsi oleh tim *user*. *Table datamart* ini bernama Push Notif CASA yang berarti digunakan untuk mengirim notifikasi atau pesan otomatis kepada KCU/KCP melalui *email*.

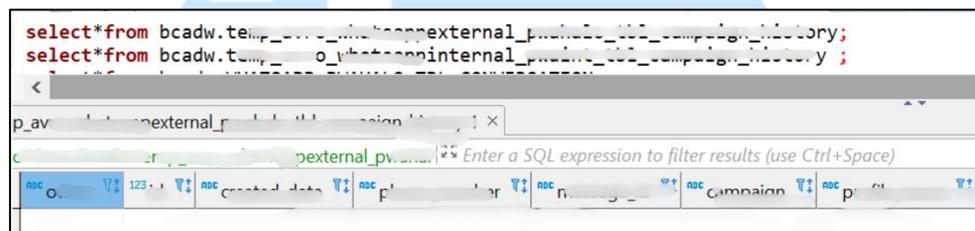
### 3.2.5 Proyek New Pricing WhatsApp

Peserta magang juga turut terlibat dalam Proyek New Pricing WhatsApp yang sebelumnya ditangani oleh Tim Channel di DWH, tetapi sekarang diambil alih kepada Tim CCF. Proyek ini akan mengubah dan memengaruhi tiga *table* utama WhatsApp yang terdapat di Data Warehouse, yakni WA Campaign History, WA Conversation, dan WA Push Notification Fact. Campaign History akan berfokus pada pencatatan interaksi pengguna atau nasabah dengan sistem kampanye pesan melalui WhatsApp BCA terutama dalam menekan simbol ‘#’ yang memicu sistem mengirimkan respon otomatis. Push Notification mencatat setiap pesan notifikasi yang dikirimkan satu arah, baik itu kampanye ataupun transaksional kepada nasabah. Selanjutnya, Conversation dalam konteks ini mengacu pada percakapan interaktif antara nasabah/pengguna dan sistem layanan WhatsApp BCA. Komunikasi ini dapat dimulai oleh sistem BCA itu sendiri ataupun oleh nasabah yang tercatat sebagai satu sesi percakapan.

Perubahan utama yang dilakukan adalah penyesuaian proses penurunan data ke Data Warehouse agar tidak menggunakan *file* lama berupa LST, melainkan sudah mengikuti standar terbaru yang diterapkan Tim DWH dengan bermigrasi menggunakan format AVRO. Adapun, *table* WA Push Notification Fact yang ingin dilakukan penambahan kolom baru. Proyek berkoordinasi dengan antar time, terutama dengan tim aplikasi bernama

Customer Touchpoint (CTS) yang menurunkan data aktivitas WhatsApp ke Tim Data Warehouse. Kemudian, Tim MIS (Management Information System) di Data Management A (DTM-A) akan menggunakan *datamart* yang telah di-*load* Tim DWH untuk membuat sebuah *report* dan analisis terkait biaya performa *channel* WhatsApp.

## 1. Source File



```

select*from bcadw.temp_..._whatsappexternal_..._campaign_history;
select*from bcadw.temp_..._whatsappinternal_..._campaign_history;

```

**Gambar 3.52** Temp AVRO WhatsApp

Gambar 3.52 menunjukkan *table* Temp AVRO sebagai *source* untuk *development* proyek ini. *Source* data tersebut diturunkan oleh tim aplikasi atau Tim Customer Touchpoint (CTS) yang telah melakukan migrasi penurunan data ke DWH menggunakan format AVRO menggantikan format lamanya, yaitu *flatfile* dengan format LST. Setiap kali tim aplikasi menurunkan data ke DWH, mereka mengirimkan dua jenis *source* sekaligus untuk masing-masing *development table* Fact, yaitu dari *database* internal dan eksternal WhatsApp. Secara data dan struktur kolom keduanya itu sama yang membedakan hanyalah *database* asalnya. Hal ini juga serupa dengan *table* Fact lainnya yang terbagi menjadi dua *source*, internal dan eksternal sehingga total *source table* Temp AVRO berjumlah enam. Proses konversi turunan data aplikasi ke *table* Temp AVRO dilakukan oleh Tim Data Standardization pada *environment* Big Data.

## 2. Development

*Development* proyek New Pricing WhatsApp berfokus pada pengembangan migrasi *source* data yang sebelumnya menggunakan format *file* LST menjadi format AVRO. Migrasi dilakukan untuk ketiga *table fact* WhatsApp, yaitu WA Push Notification, WA Campaign History, dan WA

Conversation Fact. Kemudian, proyek Pricing WhatsApp yang sudah *existing* sebelumnya ini juga ditambahkan beberapa kolom baru di *datamart* WA Push Notification Fact.

a. Penambahan Kolom Baru

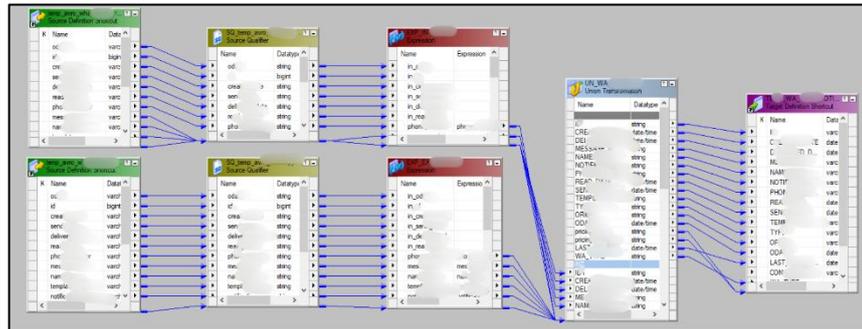
```
ALTER TABLE fact.WA_PUSH_NOTIFICATION_FACT ADD (
  FLAG_CHANNEL VARCHAR2 (50),
  CHANNEL_NAME VARCHAR2 (255),
  ADDL_INFO VARCHAR2 (255),
  CATEGORY_PRICE VARCHAR2 (32),
  CATEGORY_TYPE VARCHAR2 (32));

ALTER TABLE fact.WA_PUSH_NOTIFICATION_FACT ADD COLUMNS (
  FLAG_CHANNEL STRING,
  CHANNEL_NAME STRING,
  ADDL_INFO STRING,
  CATEGORY_PRICE STRING,
  CATEGORY_TYPE STRING);
```

Gambar 3.53 ALTER TABLE WA Push Notification Fact

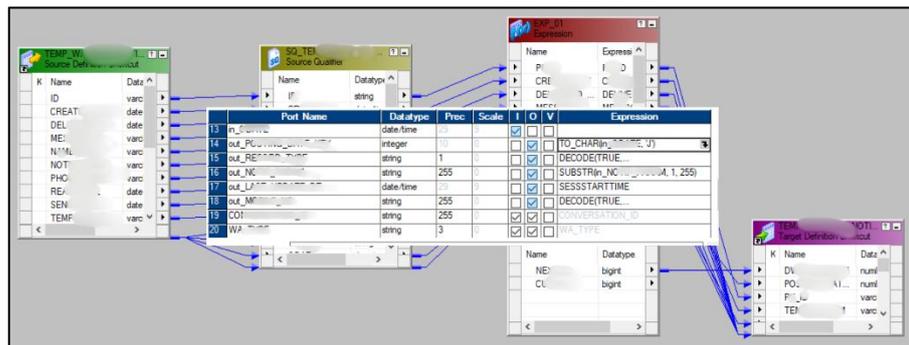
Hanya *table* WA Push Notification yang perlu dilakukan penyesuaian struktur *table* dengan adanya DDL (Data Definition Language) penambahan lima kolom baru. Kolom tersebut berupa FLAG CHANNEL NAME, ADDL INFO, CATEGORY PRICE, dan CATEGORY TYPE. Pada Oracle, kelima kolom tersebut ditambahkan pada tiga *table staging*, yaitu Temp 1, Temp 2, dan Fact. Sementara di Big Data, penambahan kolom hanya dilakukan pada *table fact* Hive yang telah di-*offload* dari Oracle.

b. Mapping Informatica PowerCenter



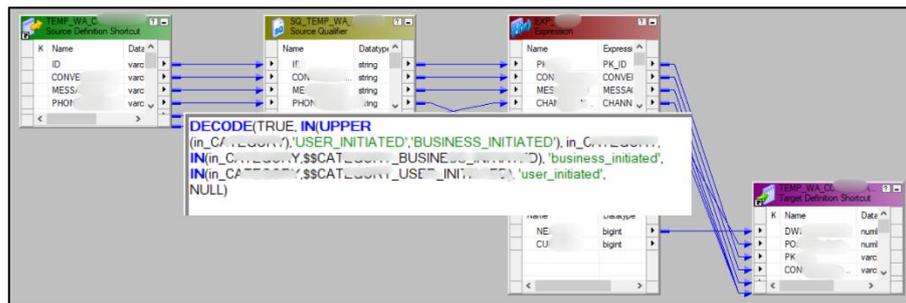
Gambar 3.54 Mapping WA Push Notif Temp 1

Disini *mapping* Temp 1 untuk ketiga *table* WhatsApp memiliki struktur alur pemrosesan yang sama seragam. Pada Gambar 3.54 peserta magang menggunakan salah satu contoh, yakni *mapping* Temp 1 WA Push Notif. Proses *mapping* diawali dengan dua *source* yang berasal dari Temp AVRO Internal dan Temp AVRO External. Seluruh kolom ditarik ke dalam transformasi Source Qualifier dan dilanjutkan dengan transformasi Expression. Disini, terdapat penambahan satu *port* atau kolom baru, yaitu WA TYPE di mana kolom tersebut digunakan sebagai penanda asal *database*. *Value* WA TYPE akan di-*hardcode* dengan berisi nilai 'INT' untuk *source* dari Internal dan 'EXT' untuk *source* dari External sehingga tetap dikenali asalnya. Kemudian, kolom LAST UPDATE DT juga ditambahkan menggunakan *syntax* SESSSTARTTIME. Selain itu, beberapa tipe data pada *source* Temp AVRO (bernilai String semua) diubah tipe datanya sesuai dengan *requirement user*. Kedua aliran data dari Internal dan External akan digabungkan melalui transformasi Union sehingga menjadi satu kesatuan *table* yang sama. Hasil dari Union langsung diarahkan ke Target Definition dengan *table* Temp 1 WA Push Notif sebagai target yang telah ditambahkan sebuah Pre-SQL TRUNCATE TABLE. Proses transformasi pada Temp 1 ini juga berlaku sama dengan Temp 1 WA Campaign History dan WA Conversation.



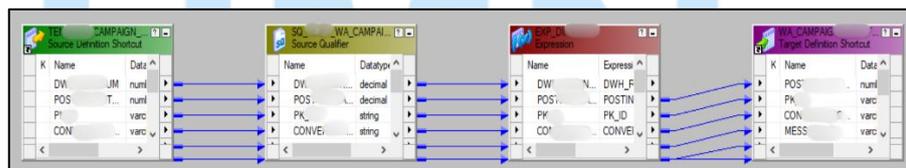
Gambar 3.55 Mapping WA Push Notif Temp 2

Proses *mapping* Temp 2 untuk WA Push Notif menggunakan target data Internal dan External yang telah digabungkan pada Temp 1 sebagai *source*. Pada *mapping* tersebut (Gambar 3.55), hanya terdapat dua transformasi yang digunakan, yaitu Expression dan Sequence Generator. Expression berfungsi untuk melakukan beberapa *logic*, termasuk mengubah ODATE (tanggal data diturunkan oleh Tim WhatsApp) menjadi POSTING DATE KEY dalam format Julian *date*. Dilakukan pula normalisasi format nomor telepon yang memiliki format tidak seragam, seperti '+62', '62', dan lain-lain sehingga disesuaikan agar konsisten dengan menggunakan awalan '08'. Kemudian, NOTIF PARAM memiliki struktur yang panjang sehingga hanya dibatasi hingga 255 karakter agar sesuai dengan batasan *length* kolom tersebut. Transformasi Sequence Generator untuk menghasilkan nomor urut otomatis yang menjadi *identifier* pada Temp 2 bernama DWHROWNUM. Terakhir, semua data dimuat ke *table* target WA Push Notif Temp 2.



**Gambar 3.56** Mapping WA Conversation Temp 2

Pada *mapping* Temp 2 untuk WA Conversation, alurnya sama persis dengan *mapping* dua *table* lainnya. Namun, yang membedakan hanyalah *logic* untuk beberapa kolom pada masing-masing *table*. Disini, terdapat tambahan *port* baru yang berfungsi sebagai *output* bernama CATEGORY NAME. *Port* tersebut diberikan *logic* untuk mengelompokkan jenis percakapan ke dalam dua kategori sesuai dengan kebutuhan *requirement*. *Logic* pada Gambar 3.56 akan mengecek apakah nilai *category* termasuk ke dalam kategori ‘business\_initiated’ atau ‘user\_initiated’ yang ditetapkan sesuai dengan parameter yang telah didefinisikan pada level *mapping*.

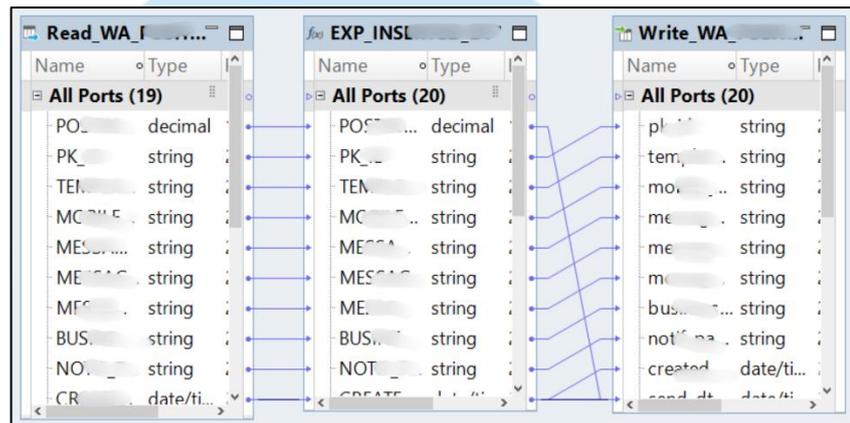


**Gambar 3.57** Mapping WA Conversation Fact

Pada *staging* terakhir, di mana data yang di-load ke *table fact* WA akan melalui alur yang sama untuk masing-masing *table fact* tanpa adanya *logic* lebih lanjut. Disini, peserta magang mengambil salah satu *mapping* untuk *staging* terakhir pada WA Conversation Fact. Hasil dari *table* Temp 2 akan menjadi *source* dalam transformasi Source Qualifier. Kemudian, data dialirkan ke Expression untuk menambahkan *port* LAST UPDATE DATE dengan *syntax* SESSSTARTTIME. Data

langsung dialirkan ke Target Definition yang akan menjadi *datamart* proyek WhatsApp. Alur transformasi pada Gambar 3.57 juga diterapkan serupa untuk kedua *table fact* lainnya.

c. Mapping Informatica Developer



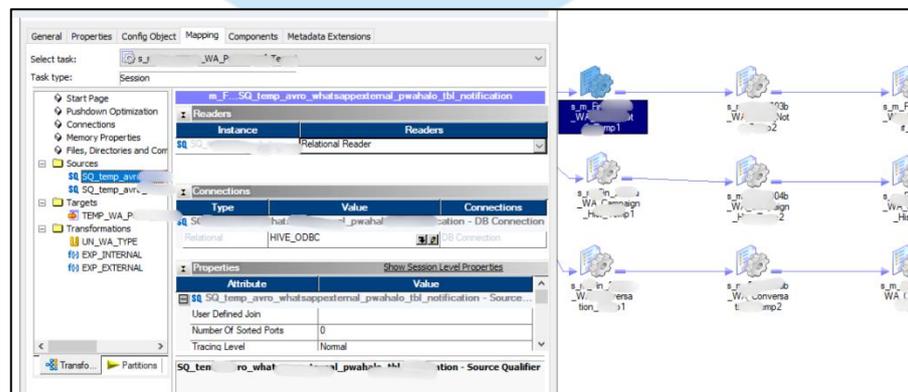
Gambar 3.58 Mapping WA Push Notification Big Data

Gambar 3.58 menunjukkan *mapping* WA Push Notification pada lingkungan Big Data. *Development* yang dilakukan ialah penyesuaian struktur kolom data dengan menambahkan *port* baru pada *source* dan *target* karena terdapat penambahan kolom juga di Big Data. Penambahan kolom ini perlu dilakukan sebelum kolom POSTING DATE KEY karena kolom tersebut berperan sebagai partisi pada *table fact* Hive Big Data sehingga tidak bisa ditambahkan setelahnya. Setelah struktur data sudah diperbarui, selanjutnya dilakukan proses *offload* dari *table fact* dari Oracle (Read) ke *table fact* di Hive Big Data (Write). Terdapat pula transformasi Expression untuk menambahkan kolom dan *logic* untuk INSERTED DATE sebagai penanda tanggal kapan data masuk ke *table* Big Data. Untuk kedua *table fact* WA lainnya juga dilakukan proses *offload* yang sama.

3. Testing Development

User Acceptance Testing proyek WhatsApp Pricing baru dimulai pada tanggal 2 Juni 2025. Selama masa *testing* ini, terdapat beberapa temuan yang perlu segera diperbaiki oleh peserta magang. Salah satunya adalah kesalahan format dan bentuk jam di beberapa kolom bertipe data *date*. Proses *testing* dilakukan secara paralel dan terkoordinasi, mulai dari Tim Data Standardization DWH menunggu *testing* yang dilakukan oleh Tim CTS (Customer Touchpoint) untuk menurunkan data UAT ke DWH yang berformat AVSC (AVRO Schema). Kemudian, dari Tim Data Standardization akan memroses data turunan ke *tabel* Temp AVRO WhatsApp Internal dan External. Setelah *source* data tersedia di DWH, barulah peserta magang melanjutkan proses ETL hingga menjadi *datamart*. *Datamart* yang ter-generate akan dilanjutkan oleh Tim MIS (Management Information System) DTM-A untuk membentuk sebuah *report*.

a. Workflow Informatica PowerCenter

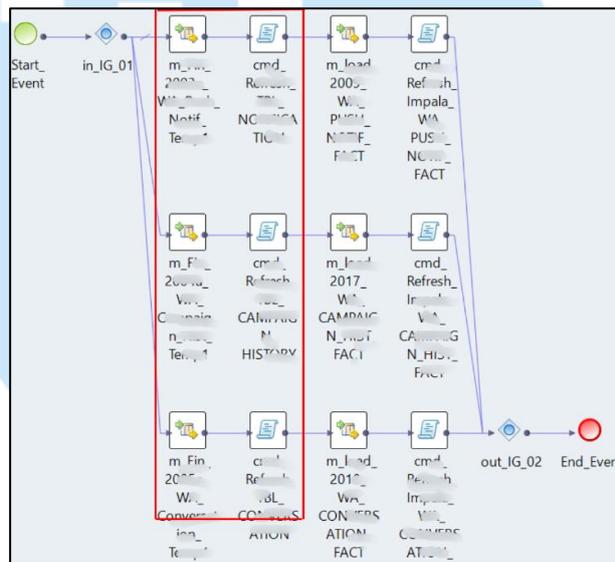


Gambar 3.59 Workflow WhatsApp Informatica PowerCenter

Pada tahap *testing*, dilakukan beberapa penyesuaian pada *workflow* WhatsApp agar proses eksekusi dapat berjalan dengan benar terhadap sumber data AVRO yang baru dimigrasi. Salah satu perubahan penting yang dilakukan ialah pergantian seluruh *session* untuk WA Fact Temp 1 melalui *setting connection* pada Source Qualifier (SQ) dari sebelumnya menggunakan *connection file* LST ke *connection* Hive ODBC. Hal tersebut dilakukan karena source data *table* Temp AVRO

WhatsApp Internal dan External kini berasal dari lingkungan Big Data (Hive) sehingga koneksi yang digunakan pun harus disesuaikan agar *mapping* atau *session* dapat membaca data *source* dengan tepat.

b. Workflow Informatica Developer



Gambar 3.60 Workflow WhatsApp Informatica Developer

Pada *workflow* WhatsApp di Informatica Developer Gambar 3.60, proses dimulai dengan memanfaatkan *inclusive gateway* untuk memberikan percabangan paralel menuju masing-masing *table fact* WhatsApp, yaitu Campaign History, Conversation, dan Push Notification. Dalam *workflow* tersebut, terdapat beberapa *object* yang sebelumnya digunakan namun kini sudah tidak relevan, ditandai dengan kotak merah tersebut. Objek-objek tersebut dihapus, termasuk beberapa *mapping* Temp 1 yang tidak digunakan lagi beserta *command refresh*-nya karena saat ini seluruh alur sudah dimigrasikan dan menggunakan format AVRO sebagai *source*, maka keberadaan objek tersebut dianggap tidak diperlukan lagi dalam *workflow* terbaru.



### 3.3 Kendala yang Ditemukan

Selama melaksanakan kegiatan kerja magang di PT Bank Central Asia Tbk sebagai Data Engineer, peserta magang juga menghadapi beberapa kendala. Walaupun seluruh proses magang didampingi oleh mentor dan berjalan sesuai arahan yang jelas, tetapi terdapat kendala teknis maupun nonteknis yang memerlukan waktu agar peserta magang dapat beradaptasi. Kendala-kendalanya ialah sebagai berikut.

1. Peserta magang perlu menyesuaikan diri secara mendalam untuk memahami konsep dan alur kerja Data Warehouse (DWH) BCA yang cukup kompleks. Hal ini mencakup *environment database* utama, yaitu Oracle dan Big Data yang masing-masing memiliki ketentuan dan struktur yang berbeda, istilah-istilah baru yang digunakan dalam konteks DWH, aturan-aturan baku dalam melakukan *development* proyek, serta penggunaan *tools* DWH.
2. Peserta magang sempat mengalami kesulitan dalam memahami struktur tim dan alur komunikasi antar divisi atau *user* lain. Dalam suatu proyek, biasanya terdapat banyak pihak yang terlibat, mulai dari Tim Business Analyst, Product Owner, Tim UAT, Tim Aplikasi, dan lain-lain yang berkoordinasi melalui *group* Microsoft Teams. Hal ini membuat munculnya tantangan dalam mengidentifikasi peran dari masing-masing anggota sehingga khawatir salah menghubungi pihak yang tidak tepat.
3. Peserta magang tidak memiliki hak akses penuh terhadap tabel-tabel tertentu, baik di Oracle maupun Big Data. Misalnya, peserta magang tidak dapat mengeksekusi perintah *query* CREATE, ALTER, SELECT, ataupun UPDATE terhadap tabel selama *development* proyek berlangsung karena keterbatasan *permission*.

### 3.4 Solusi atas Kendala yang Ditemukan

Dalam menghadapi beberapa kendala selama pelaksanaan magang di PT Bank Central Asia Tbk, peserta magang tidak hanya berfokus pada kendala yang

dihadapi, tetapi juga menemukan solusi untuk mengatasinya. Berikut solusi-solusi yang ditemukan.

1. Sejak awal masuk kerja magang, peserta magang diberikan materi dalam bentuk teori, bacaan, serta sesi *sharing knowledge* oleh mentor Tim CCF. Materi mencakup pemahaman konsep dan alur kerja di Data Warehouse BCA, *tools* yang digunakan, dan lain-lain. Peserta magang perlu memahami secara mendalam dengan mencatat setiap materi yang diberikan agar terus diingat dan dijadikan referensi. Kemudian, hal tersebut diimbangi dengan eksplorasi terhadap *mapping-mapping* yang sudah ada di *repository* DWH.
2. Peserta magang aktif bertanya kepada mentor mengenai struktur peran *user-user* selama proyek berlangsung. Selain itu, peserta magang lebih fokus dan memperhatikan secara detail saat *meeting* proyek karena dari pembahasan yang disampaikan, biasanya cukup jelas terlihat siapa yang memegang peran-peran tertentu di dalam proyek.
3. Mentor dan rekan satu tim turut membantu dengan memberikan *grant access* tabel tertentu kepada peserta magang. Akses yang diberikan umumnya adalah SELECT agar peserta magang dapat melakukan pengecekan terhadap data. Apabila peserta magang membutuhkan perubahan, seperti CREATE, ALTER, UPDATE, DELETE, dan lain-lain, maka peserta magang akan menulis dan mengirimkan *query* tersebut melalui Microsoft Teams kepada mentor atau rekan satu tim untuk dieksekusi.