

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Posisi Data Engineer di PT Bank Central Asia Tbk terletak dalam biro Data Management B (DTM-B) dengan grup GSIT (Group Strategic Information Technology). Peserta magang berada dalam tim History Development khususnya bagian peminjaman (*Lending*). Tim History Development merupakan tim yang bertugas untuk mengelola seluruh data PT Bank Central Asia Tbk mulai dari periode yang lama sampai periode yang paling baru. Pengelolaan data dilakukan dengan metode ETL (Extract, Transform, Load) yang bertujuan untuk memenuhi kebutuhan user seperti menyediakan data yang diperlukan oleh user, memproses data user untuk keperluan analisis dan menyimpan data ke dalam database PT Bank Central Asia Tbk.



Gambar 3. 1 Anggota Tim Lending

Gambar 3.1 merupakan anggota dari tim *Lending*. Proses koordinasi yang berlangsung dalam tim *History Development* khususnya pada bagian peminjaman (*Lending*) dilakukan dengan leader tim *Lending* yaitu Bapak Sandi Wijaya dan

dengan rekan satu tim Lending. Koordinasi proyek dan kinerja tim dipimpin oleh Bapak Sandi Wijaya mulai dari penentuan PIC setiap proyek yang masuk, pembagian tugas, dan memimpin meeting yang dilakukan secara internal maupun eksternal. Koordinasi bersama dengan Bapak Sandi Wijaya selalu dilakukan secara online melalui Microsoft Teams karena penempatan site kantor yang berbeda dan koordinasi tim dapat dilakukan secara offline dengan tim yang berada di kantor yang sama.

Tim History Development khususnya dalam bidang peminjaman (Lending) memiliki prosedur alur kerja yang akan diterapkan dalam setiap proyek. Alur kerja tersebut terbagi menjadi tiga alur yaitu sharing requirement, development dan testing. Sharing requirement merupakan tahap dimana user akan melakukan meeting untuk menjelaskan kebutuhan dari proyek tersebut. Requirement tersebut dituliskan dalam bentuk sebuah tabel dalam website internal PT Bank Central Asia Tbk yang bernama blueprint. Meeting sharing requirement biasanya akan dihadiri oleh Bapak Sandi Wijaya atau PIC dari masing-masing proyek serta teman dalam satu tim yang terlibat.

Tahap yang selanjutnya setelah tahap sharing requirement yaitu tahap development. Tahap development merupakan tahap dimana developer akan mulai mengerjakan proyek sesuai dengan requirement dan kebutuhan user yang telah dibahas pada saat sharing requirement. Pada umumnya proyek yang dikerjakan merupakan proyek data warehouse sehingga tools yang paling banyak digunakan untuk melakukan tahap development yaitu Informatica, Oracle, MobaXTerm dan DBeaver. Lamanya proses development ditentukan melalui tingkat kesulitan proyek dan requirement dari user. Proses development akan selalu dibahas melalui meeting mingguan yang dilakukan setiap hari Senin. Pada hari Kamis, meeting problem biasanya dilakukan untuk membahas kendala-kendala yang ditemukan pada saat proses development suatu proyek.

Tahap terakhir yang dilakukan dalam alur kerja tim Lending setelah menyelesaikan tahap development yaitu tahap UAT (User Acceptance Testing). PT

Bank Central Asia Tbk memiliki 1 tim yang dikhususkan untuk membantu melakukan tahap UAT (User Acceptance Testing) yaitu tim APP & User Acceptance Test atau biasa disebut dengan KLA. Tim UAT (User Acceptance Testing) tersebut memiliki tugas untuk membantu developer dan user melakukan verifikasi terhadap proyek yang telah selesai dikerjakan. Verifikasi dilakukan dengan berbagai macam kemungkinan skenario yang dapat terjadi dengan data ketika akan dimasukkan ke dalam database PT Bank Central Asia Tbk. Ketika tahap UAT (User Acceptance Testing) selesai dilakukan maka akan diimplementasikan kedalam database, tetapi tahap implementasi ini tidak dapat dilakukan oleh peserta magang karena peserta magang tidak memiliki akses ke dalam lingkungan production.

3.2 Tugas dan Uraian Kerja Magang

Selama kegiatan magang dilakukan, peserta magang diberikan beberapa proyek untuk dikerjakan. Dalam durasi magang selama 5 bulan, mulai dari tanggal 7 Agustus 2023 sampai dengan 6 Januari 2023 terdapat 8 proyek yang telah diselesaikan. Prosedur pengerjaan proyek tersebut dilakukan secara bertahap yaitu development kemudian dilanjutkan dengan testing. Peserta magang tidak dapat melanjutkan sampai dengan tahap implementasi dikarenakan tidak memiliki akses terhadap server production. Rincian dari kegiatan proyek selama periode magang berlangsung yaitu sebagai berikut:

Tabel 3. 1 Timeline Detail Magang

Nomor	Judul Aktivitas	Mulai	Selesai
	Datamart Data Kredit		
1	Proyek RAKC		
	Melakukan development proyek RAKC	7 Agustus 2023	9 Agustus 2023
	Melakukan testing terhadap proyek RAKC	10 Agustus 2023	11 Agustus 2023
2	Proyek Plafond Efektif Monthly		
	Melakukan development proyek Plafond Efektif Monthly	14 Agustus 2023	18 Agustus 2023
	Melakukan testing terhadap proyek Plafond Efektif Monthly	21 Agustus 2023	22 Agustus 2023

Nomor	Judul Aktivitas	Mulai	Selesai
3	Proyek Deleted Rowkey HBASE		
	Melakukan development proyek Deleted Rowkey HBASE	23 Agustus 2023	5 September 2023
	Melakukan testing terhadap proyek Deleted Rowkey HBASE	6 September 2023	8 September 2023
4	Proyek Avg 12 Mth Pricing		
	Melakukan development proyek Avg 12 Mth Pricing	11 September 2023	23 Oktober 2023
	Melakukan testing terhadap proyek Avg 12 Mth Pricing	15 September	19 September 2023
	Datamart Kartu Kredit		
5	Proyek Pricing Active & History Product BDC		
	Melakukan development proyek Active & History Product BDC	20 September 2023	13 Oktober 2023
	Melakukan testing terhadap proyek Active & History Product BDC	16 Oktober 2023	25 Oktober 2023
6	Proyek Belief		
	Melakukan development proyek Belief	6 November 2023	10 November 2023
	Melakukan testing terhadap proyek Belief	20 November 2023	24 November 2023
7	Proyek NPWP 16 Digit		
	Melakukan development Proyek NPWP 16 Digit	26 Oktober 2023	1 November 2023
	Melakukan testing terhadap proyek NPWP 16 Digit	30 Oktober 2023	27 November 2023
8	Proyek CCOS QUAL QRIT		
	Melakukan <i>development proyek</i> CCOS QUAL QRIT	13 November 2023	17 November 2023
	Melakukan <i>testing terhadap proyek</i> CCOS QUAL QRIT	20 November 2023	22 November 2023

Dalam pengerjaan proyek-proyek yang diberikan selama periode magang berlangsung. Terdapat beberapa *tools* yang digunakan untuk mengerjakan proyek tersebut. *Tools* yang digunakan yaitu tools ETL (*Extract, Transform, Load*), database, big data dan *tools* untuk mengakses server. Berikut merupakan penjelasan dari tools-tools tersebut:

- **Informatica**



Gambar 3. 2 Logo Informatica

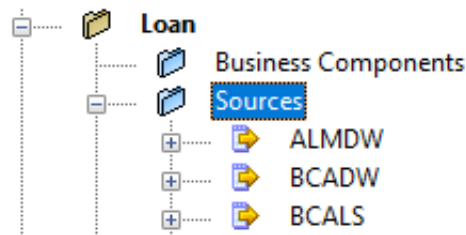
Sumber : google.com

Informatica merupakan salah satu software yang digunakan oleh PT Bank Central Asia Tbk untuk melakukan proses ETL (*Extract, Transform, Load*). Informatica memiliki empat aplikasi yang memiliki fungsi berbeda-beda yaitu Informatica PowerCenter Designer, Informatica PowerCenter Workflow, Informatica PowerCenter Monitoring dan Informatica Developer. Keempat aplikasi tersebut di dalamnya memiliki komponen-komponennya sendiri untuk membantu *user* dalam mengolah data sampai data masuk ke dalam database PT Bank Central Asia Tbk. Keempat aplikasi tersebut memegang peran yang penting untuk membentuk *Data Warehouse* yang efektif dan efisien untuk kebutuhan data.

A. Informatica PowerCenter Designer

Informatica PowerCenter Designer merupakan aplikasi yang dapat dikatakan aplikasi inti dari Informatica. Informatica PowerCenter Designer memiliki fungsi untuk melakukan proses utama dari *Data Warehouse* yaitu ETL (*Extract, Transform, Load*). Informatica PowerCenter Designer digunakan untuk membuat mapping yang terdapat banyak komponen didalamnya. Terdapat 3 komponen utama yaitu sumber data (*source*), komponen transformasi dan target.

- **Sumber Data (*Source*)**



Gambar 3. 3 Sumber Data

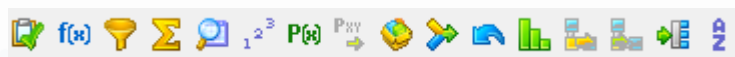
Gambar 3.3 merupakan salah satu komponen utama dalam pembuatan mapping yaitu sumber data (*source*). Komponen sumber data (*source*) dapat berbentuk file atau tabel yang akan digunakan untuk menjadi sumber pengambilan data untuk diolah di dalam mapping yang akan dibuat.

K	Name	Datatype	Length/Precision
	NEWFIELD	string	10
	NEWFIELD1	string	10
	NEWFIELD2	string	10
	NEWFIELD3	string	10
	NEWFIELD4	string	10

Gambar 3. 4 Isi Sumber Data

Gambar 3.4 merupakan isi dari komponen sumber data (*source*). Komponen sumber data (*source*) ditandai dengan warna hijau dan selalu memiliki nama kolom, tipe data kolom dan panjang karakter dari kolom tersebut.

- **Transformasi**

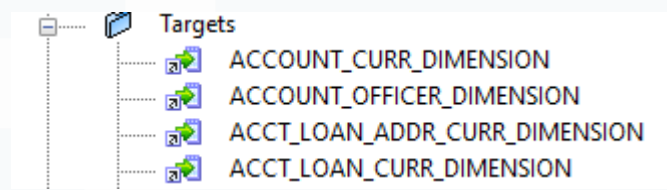


Gambar 3. 5 Bar Komponen

Gambar 3.5 merupakan kumpulan dari komponen transformasi yang berada di dalam aplikasi Informatica PowerCenter Designer. Komponen transformasi berfungsi untuk

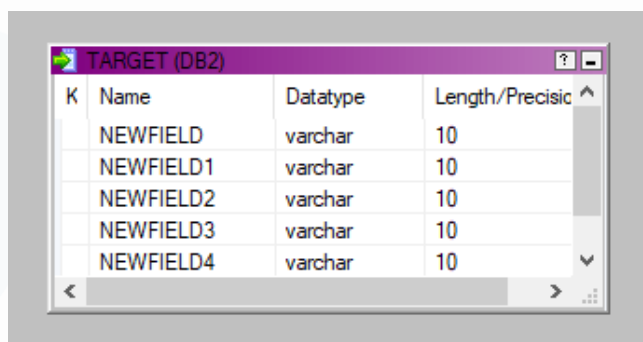
melakukan proses pengolahan data seperti filter data, penggabungan data, melakukan perhitungan dan melakukan pengurutan data. Terdapat beberapa komponen transformasi yang umumnya digunakan dalam proses ETL (Extract, Transform, Load) yaitu seperti *joiner*, *sorter*, *union*, *filter*, *expression* dan *aggregator*.

- **Target**



Gambar 3. 6 Target

Gambar 3.6 merupakan komponen target yang berisikan file atau tabel yang sudah terhubung dengan database. Komponen ini merupakan komponen yang menjadi tujuan akhir data yang diolah dalam suatu mapping. Ketika ditarik kedalam komponen target, maka data akan masuk ke dalam tabel dan tersimpan ke dalam database.



K	Name	Datatype	Length/Precisic
	NEWFIELD	varchar	10
	NEWFIELD1	varchar	10
	NEWFIELD2	varchar	10
	NEWFIELD3	varchar	10
	NEWFIELD4	varchar	10

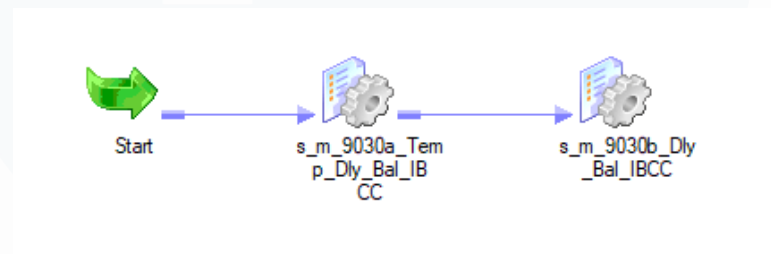
Gambar 3. 7 Isi Target

Gambar 3.7 merupakan isi dari komponen target. Komponen target merupakan tabel yang telah dibentuk di database Oracle. Komponen target ditandai dengan warna ungu dan selalu memiliki

nama kolom, tipe data kolom dan panjang karakter dari kolom tersebut.

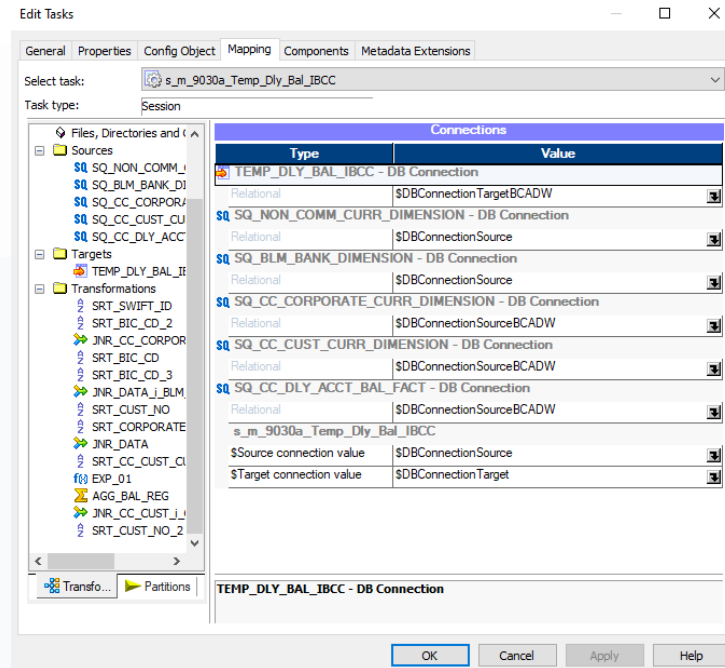
B. Informatica PowerCenter Workflow

Informatica PowerCenter Workflow merupakan aplikasi yang digunakan untuk menjalankan mapping yang telah dibuat menggunakan Informatica PowerCenter Designer. Informatica PowerCenter Workflow memudahkan user untuk menjalankan mapping sesuai dengan urutan karena dalam aplikasi ini, user dapat membentuk alur kerja yang kemudian dapat dijalankan secara otomatis.



Gambar 3. 8 Alur Kerja

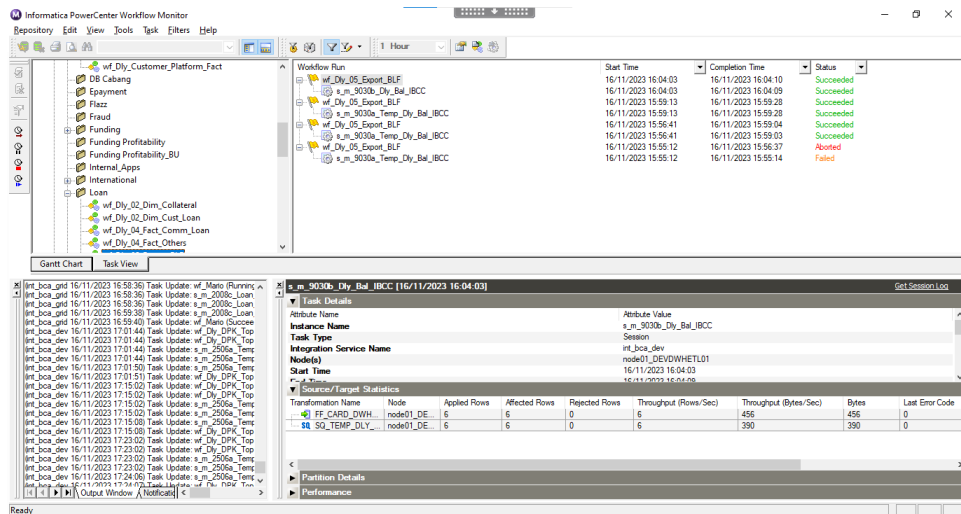
Gambar 3.8 merupakan contoh alur kerja yang berada dalam aplikasi Informatica PowerCenter Workflow. Alur kerja tersebut disimpan kedalam folder yang berbeda sesuai dengan koneksi database. Untuk menjalankan suatu mapping di dalam alur kerja Informatica PowerCenter Workflow, mapping harus dimasukkan ke dalam alur kerja tersebut dalam bentuk *session*. *Session* adalah komponen yang menghubungkan mapping menggunakan koneksi yang sesuai dengan database.



Gambar 3. 9 Menu Setting Session

C. Informatica PowerCenter Monitoring

Aplikasi Informatica PowerCenter Monitoring berfungsi untuk memonitor proses yang dilakukan oleh *workflow*. Informatica PowerCenter Monitoring menyediakan informasi mengenai proses yang sedang berjalan seperti kecepatan proses data. Informatica PowerCenter Monitoring akan memberikan *error log* apabila terdapat kesalahan pada saat melakukan proses data, hal ini sangat membantu user untuk memperbaiki *error* tersebut karena penyebab dari *error* tersebut akan disebutkan dalam *error log*.

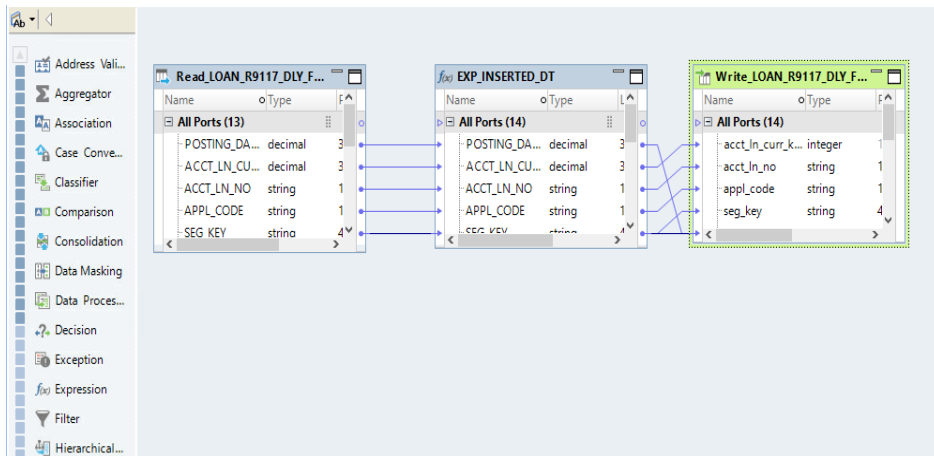


Gambar 3. 10 Tampilan Informatica PoweCenter Monitoring

Gambar 3.10 merupakan tampilan dari aplikasi Informatica PowerCenter Monitoring. Terdapat berbagai macam informasi seperti status dari proses dan jumlah data yang telah terproses.

D. Informatica Developer

Informatica developer memiliki peran yang sama seperti aplikasi Informatica Powercenter. Perbedaan Informatica Developer dengan Informatica Powercenter yaitu dari segi fitur dan aplikasi. Informatica PowerCenter memiliki 3 aplikasi terpisah untuk melakukan proses ETL (*Extract, Transform, Load*) sedangkan Informatica Developer hanya perlu menggunakan satu aplikasi yang sudah terdapat *designer, workflow* dan *monitoring* di dalamnya. PT Bank Central Asia Tbk menggunakan Informatica Developer untuk melakukan *backup* dan *offload* data dari database Oracle ke HIVE bigdata. Informatica Developer menggunakan koneksi HADOOP untuk menghubungkan ke dalam bigdata. Selain tabel *offload*, Informatica Developer juga digunakan untuk membuat tabel RAW yang dimana dikhususkan untuk menyimpan data mentah yang belum diolah. *Runtime* yang digunakan oleh Informatica Developer yaitu Blaze dan Spark.



Gambar 3. 11 Mapping Informatica Developer

Gambar 3.11 merupakan contoh mapping yang dibuat dengan Informatica Developer. Informatica Developer kurang lebih memiliki tampilan dan fungsi yang mirip dengan Informatica PowerCenter, hanya saja Informatica Developer memiliki semua fiturnya dalam satu aplikasi saja.

- **Oracle**



Gambar 3. 12 Logo Oracle

Sumber : google.com

Oracle merupakan *software* yang berfungsi untuk membuat *relational database management system (RDBMS)*. *Structured Query Language (SQL)* merupakan bahasa pemrograman yang digunakan oleh aplikasi Oracle. Fungsi utama aplikasi Oracle yaitu untuk memproses bentuk data yang masih mentah ke dalam suatu tabel yang telah dibuat di dalam database sehingga data dapat tersimpan dengan rapi sesuai dengan struktur tabel dalam database. Structured query language dibagi menjadi 4 yaitu *Data Definition Language (DDL)*, *Data Manipulation Language*

(DML), *Data Control Language (DCL)* dan *Transaction Control Language (TCL)* [4].

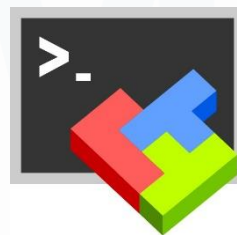
- **DBeaver**



Gambar 3. 13 Logo Dbeaver
Sumber : google.com

DBeaver merupakan software yang memiliki fungsi sama dengan Oracle yaitu untuk membuat *relational database management system (RDBMS)*. DBeaver memiliki kompatibilitas terhadap beberapa database yang menggunakan bahasa pemrograman *Structured Query Language (SQL)* seperti MySQL dan PostgreSQL. PT Bank Central Asia Tbk menggunakan DBeaver untuk melakukan load data ke dalam server big data dengan menggunakan koneksi HADOOP.

- **MobaXterm**



Gambar 3. 14 Logo MobaXterm
Sumber : google.com

MobaXterm merupakan salah satu aplikasi *Secure Shell* yang berbasis dengan bahasa pemrograman Linux. MobaXterm digunakan untuk melakukan *remote computing* agar dapat mengakses server PT Bank Central Asia Tbk secara jarak jauh. MobaXterm biasanya digunakan untuk

mengambil file yang diperlukan pada saat melakukan development atau memasukkan file kedalam server sehingga aplikasi lain dapat membaca file ke dalam lokasi folder yang telah ditentukan. MobaXterm juga digunakan untuk memberikan file kepada tim aplikasi atau tim yang lain [4].

3.2.1 *Proyek RAKC*

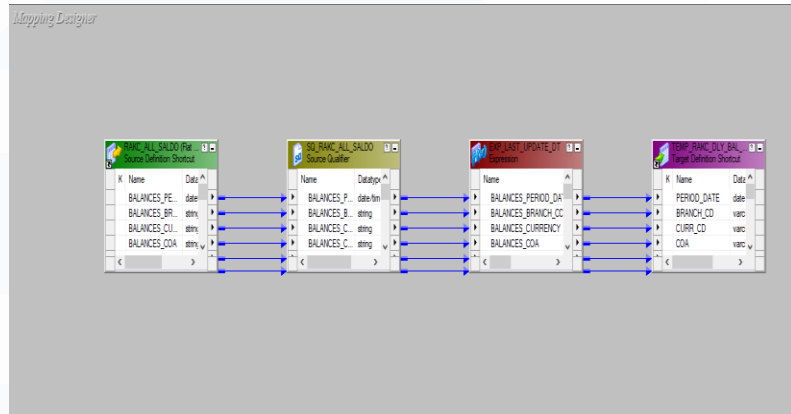
A. Development

Proyek RAKC merupakan proyek memasukkan data dari tim aplikasi ke dalam database data warehouse PT Bank Central Asia Tbk. Data dikirimkan dengan bentuk *flat file* kemudian data diolah dan digabungkan dengan kolom yang dibutuhkan menjadi satu kesatuan dalam tabel database. Data yang ingin dimasukkan ke dalam tabel RAKC yang baru yaitu data transaksi di level cabang, terdapat beberapa logic untuk perhitungan data transaksi tersebut. Proyek RAKC terdiri dari tiga tabel utama yaitu RAKC_DLY_BAL_FACT, RAKC_IDR_DLY_TRANS_FACT dan RACK_VALAS_DLY_TRANS_FACT. Tabel-tabel tersebut terpecah menjadi dua tabel yaitu tabel *temporary* dan tabel *fact*. Tabel *temporary* adalah tabel sementara yang digunakan untuk menampung data agar data dapat diolah dan akan dilakukan TRUNCATE setiap kali sebelum proses dijalankan untuk mengosongkan tabel, sedangkan tabel *fact* merupakan tabel yang berisikan data-data transaksi. Berikut merupakan penjelasan dan proses ETL (*Extract, Transform, Load*) dari proyek RAKC.

▪ **RAKC_DLY_BAL_FACT**

Tahap pertama dalam melakukan development tabel RAKC_DLY_BAL_FACT yaitu membuat tabel TEMP_1 untuk memasukkan data yang masih mentah dari *flat file* ke dalam tabel *database*, sehingga dapat dilakukan proses *transform*. Flat file diberikan oleh tim aplikasi ke tim data

warehouse. Proses ini dilakukan dengan menggunakan aplikasi Informatica PowerCenter Designer.



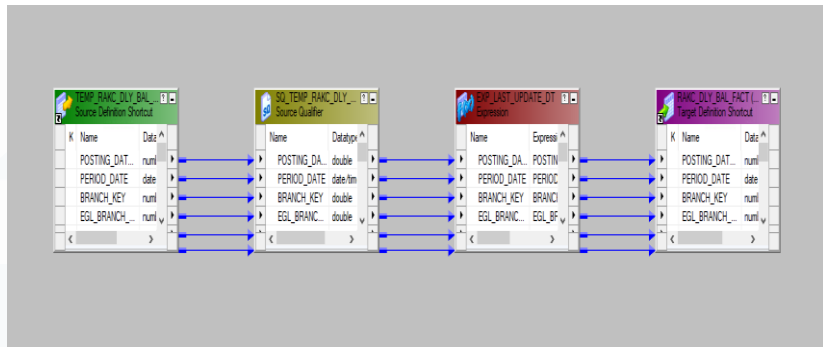
Gambar 3.15 Mapping TEMP 1 RAKC_DLY_BAL_FACT

Gambar 3.15 merupakan mapping TEMP_1 dari tabel RAKC_DLY_BAL_FACT. Proses yang dilakukan yaitu memindahkan data mentah dari flat file yang dikirimkan oleh tim aplikasi ke dalam tabel database Oracle. Data mentah akan dimasukkan ke dalam source qualifier yang berfungsi untuk melakukan seleksi data dan kolom yang diperlukan. Proses berikutnya yang dilakukan yaitu memasukkan data dan kolom yang sudah diseleksi ke dalam transformasi expression untuk ditambahkan *logic* LAST_UPDATE_DT yang memiliki fungsi untuk mengeluarkan tanggal dan waktu secara otomatis pada saat data diproses. Kemudian proses yang terakhir yaitu memasukkan data-data yang sudah terseleksi ke dalam tabel database Oracle.



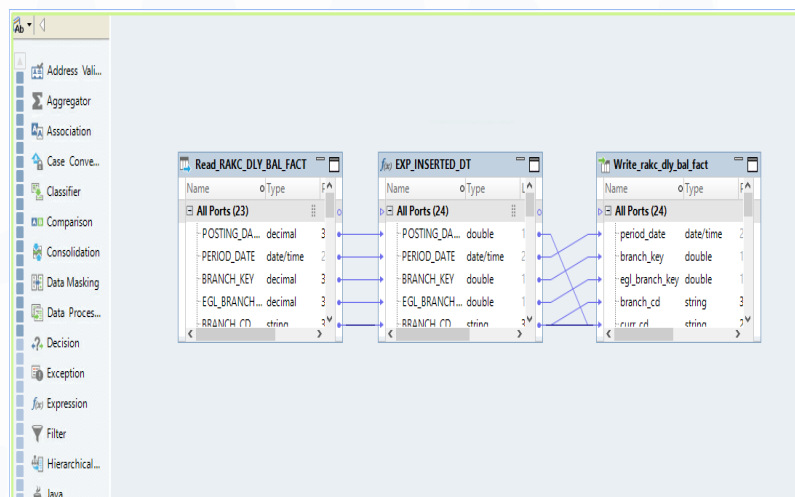
Gambar 3. 16 Mapping TEMP 2 RAKC_DLY_BAL_FACT

Gambar 3.16 merupakan proses mapping TEMP_2 dari tabel RAKC_DLY_BAL_FACT. Dalam mapping tersebut, dilakukan proses *transform* terhadap data yang sudah diseleksi sebelumnya di mapping TEMP_1. Proses ini dilakukan untuk mengambil kolom KEY yang cocok dari beberapa tabel yang dibutuhkan oleh user. Proses pengambilan kolom tersebut dilakukan dengan menggunakan transformasi LOOKUP yang berfungsi untuk menggabungkan data yang memiliki kondisi cocok. Untuk melakukan proses LOOKUP tersebut data dari TEMP_1 digabungkan dengan kolom dummy yang berisi huruf X di dalamnya. Setelah semua kolom yang digunakan telah didapatkan, maka data tersebut akan ditambahkan logic yang ingin diterapkan, pada proyek ini logic yang diterapkan adalah logic perhitungan dengan kondisi yang telah ditentukan. Langkah yang terakhir yaitu menarik semua kolom yang telah dilakukan transformasi ke dalam tabel database Oracle.



Gambar 3.17 Mapping FACT RAKC_DLY_BAL_FACT

Gambar 3.17 merupakan proses terakhir dari pembentukan tabel RAKC_DLY_BAL_FACT. Proses yang dilakukan yaitu proses load dengan memasukkan data yang telah melalui proses *extract* dan *transform*. Sebelum data dimasukkan ke dalam target tabel akhir yaitu RAKC_DLY_BAL_FACT, dilakukan penambahan transformasi expression untuk menambahkan kolom LAST_UPDATE_DT yang berfungsi untuk mengeluarkan tanggal dan waktu secara otomatis pada saat data diproses. Kemudian langkah yang terakhir yaitu melakukan *push* data ke dalam tabel target akhir di database Oracle.



Gambar 3.18 Mapping Offload RACK_DLY_BAL_FACT

Gambar 3.18 merupakan mapping yang melakukan proses offload data RAKC_DLY_BAL_FACT. Proses offload dilakukan dengan menggunakan aplikasi Informatica Developer. Proses offload ini berfungsi untuk memasukkan data yang berada dalam lingkup database oracle ke dalam lingkup big data dengan menggunakan DBeaver. Tabel *read* merupakan tabel yang bersumber dari tabel oracle. Setelah tabel dari oracle selesai dibaca, akan ditambahkan transformasi expression untuk menambahkan kolom INSERTED_DT. Kolom INSERTED_DT berfungsi untuk memberikan informasi tanggal berdasarkan partisi offload data tersebut dilakukan. Proses yang terakhir dilakukan setelah diproses yaitu melakukan *write* ke dalam tabel big data RAKC_DLY_BAL_FACT.

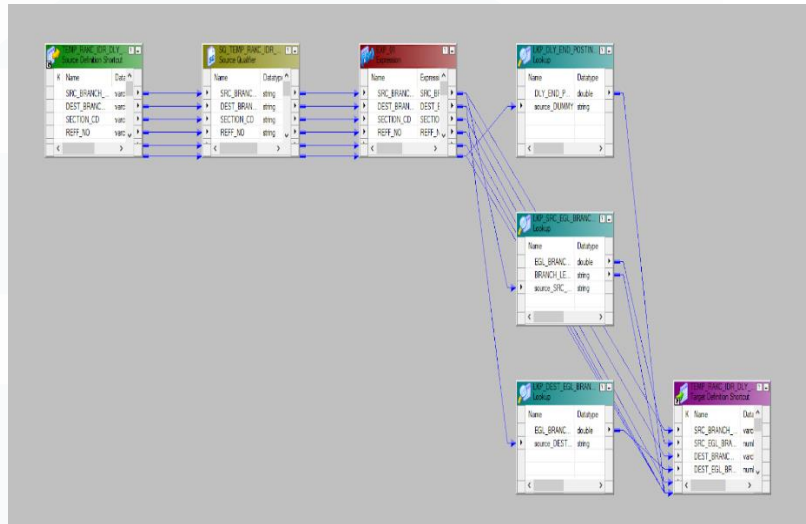
- **RAKC_DLY_IDR_TRANS_FACT**



Gambar 3. 19 Mapping TEMP 1 RAKC_DLY_IDR_TRANS_FACT

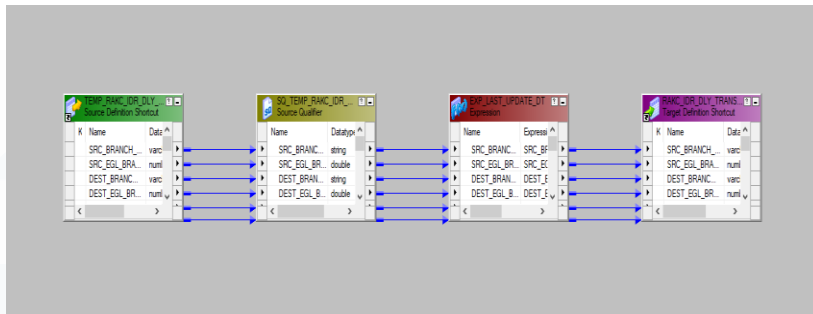
Gambar 3.19 merupakan tahap development yang pertama yaitu mapping TEMP_1 dari tabel RAKC_DLY_IDR_TRANS_FACT. Tabel ini memiliki isi yang mirip dengan RAKC_DLY_BAL_FACT, hanya saja dalam tabel ini transaksi yang diambil hanya transaksi yang dilakukan dengan kurs rupiah. Proses yang dilakukan dalam TEMP_1 yaitu membaca sumber data dari file yang diberikan oleh tim aplikasi dan memasukkannya ke dalam tabel TEMP_1.

Data cleansing dan penambahan kolom LAST_UPDATE_DT dilakukan pada tahap ini.



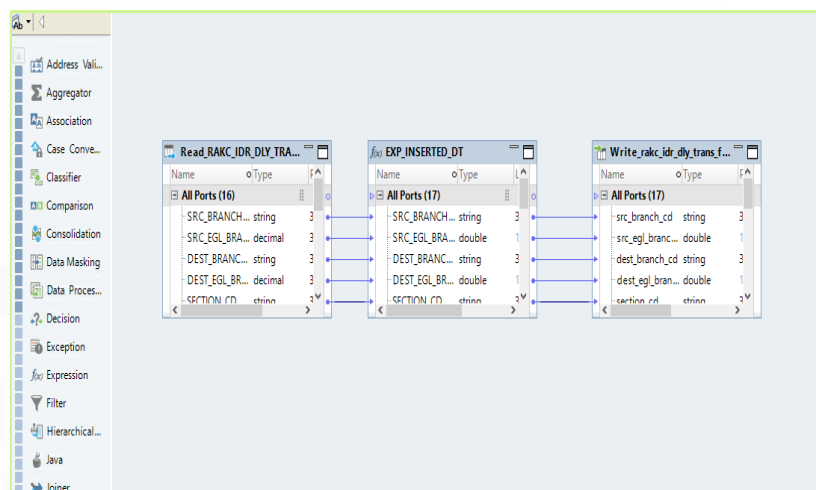
Gambar 3. 20 Mapping TEMP 2 RAKC_DLY_IDR_TRANS_FACT

Gambar 3.20 merupakan proses mapping TEMP_2 dari tabel RAKC_DLY_IDR_TRANS_FACT. Dalam mapping tersebut, dilakukan proses *transform* terhadap data yang sudah diseleksi sebelumnya di mapping TEMP_1. Proses ini dilakukan untuk mengambil kolom KEY yang cocok dari beberapa tabel yang dibutuhkan oleh user. Proses pengambilan kolom tersebut dilakukan dengan menggunakan transformasi LOOKUP yang berfungsi untuk menggabungkan data yang memiliki kondisi cocok. Untuk melakukan proses LOOKUP tersebut data dari TEMP_1 digabungkan dengan kolom dummy yang berisi huruf X di dalamnya. Setelah semua kolom yang digunakan telah didapatkan, maka data tersebut akan ditambahkan logic yang ingin diterapkan, pada proyek ini logic yang diterapkan adalah logic perhitungan dengan kondisi yang telah ditentukan. Langkah yang terakhir yaitu menarik semua kolom yang telah dilakukan transformasi ke dalam tabel database Oracle.



Gambar 3. 21 Mapping FACT RAKC_DLY_IDR_TRANS_FACT

Gambar 3.21 merupakan proses terakhir dari pembentukan tabel RAKC_DLY_IDR_TRANS_FACT. Proses yang dilakukan yaitu proses load dengan memasukkan data yang telah melalui proses *extract* dan *transform*. Sebelum data dimasukkan ke dalam target tabel akhir yaitu RAKC_DLY_IDR_TRANS_FACT, dilakukan penambahan transformasi *expression* untuk menambahkan kolom LAST_UPDATE_DT yang berfungsi untuk mengeluarkan tanggal dan waktu secara otomatis pada saat data diproses. Kemudian langkah yang terakhir yaitu melakukan *push* data ke dalam tabel target akhir di database Oracle.



Gambar 3. 22 Mapping Offload RAKC_DLY_IDR_TRANS_FACT

Gambar 3.22 merupakan mapping yang melakukan proses offload data RAKC_DLY_IDR_TRANS_FACT. Proses offload dilakukan dengan menggunakan aplikasi Informatica Developer. Proses offload ini berfungsi untuk memasukkan data yang berada dalam lingkup database oracle ke dalam lingkup big data dengan menggunakan DBeaver. Tabel *read* merupakan tabel yang bersumber dari tabel oracle. Kolom INSERTED_DT berfungsi untuk memberikan informasi tanggal berdasarkan partisi offload data tersebut dilakukan. Proses yang terakhir dilakukan setelah diproses yaitu melakukan *write* ke dalam tabel big data RAKC_DLY_IDR_TRANS_FACT.

- **RAKC_DLY_VALAS_TRANS_FACT**



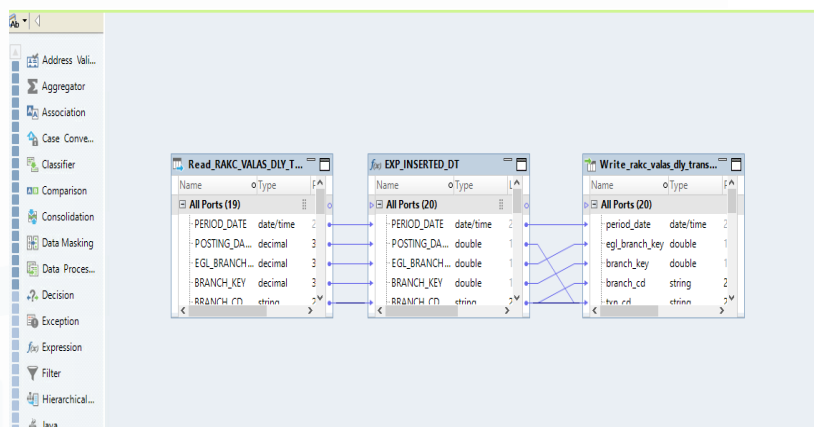
Gambar 3. 23 Mapping TEMP 1 RAKC_DLY_VALAS_TRANS_FACT

Gambar 3.23 merupakan tahap development yang pertama yaitu mapping TEMP_1 dari tabel RAKC_DLY_VALAS_TRANS_FACT. Tabel ini memiliki isi yang mirip dengan RAKC_DLY_VALAS_TRANS_FACT, hanya saja dalam tabel ini transaksi yang diambil hanya transaksi yang dilakukan dengan kurs valas atau mata uang asing. Proses yang dilakukan dalam TEMP_1 yaitu membaca sumber data dari file yang diberikan oleh tim aplikasi dan memasukkannya ke dalam tabel TEMP_1. Data cleansing dan penambahan kolom LAST_UPDATE_DT dilakukan pada tahap ini.



Gambar 3. 25 Mapping FACT RAKC_DLY_VALAS_TRANS_FACT

Gambar 3.25 merupakan proses terakhir dari pembentukan tabel RAKC_DLY_VALAS_TRANS_FACT. Proses yang dilakukan yaitu proses load dengan memasukkan data yang telah melalui proses *extract* dan *transform*. Sebelum data dimasukkan ke dalam target tabel akhir yaitu RAKC_DLY_VALAS_TRANS_FACT, dilakukan penambahan transformasi *expression* untuk menambahkan kolom LAST_UPDATE_DT yang berfungsi untuk mengeluarkan tanggal dan waktu secara otomatis pada saat data diproses. Kemudian langkah yang terakhir yaitu melakukan *push* data ke dalam tabel target akhir di database Oracle.



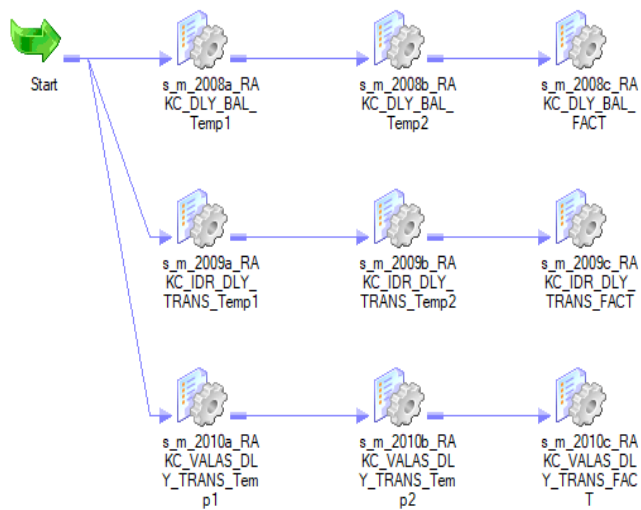
Gambar 3. 26 Mapping Offload RAKC_DLY_VALAS_TRANS_FACT

Gambar 3.26 merupakan mapping yang melakukan proses offload data RAKC_DLY_VALAS_TRANS_FACT. Proses offload dilakukan dengan menggunakan aplikasi

Informatica Developer. Proses offload ini berfungsi untuk memasukkan data yang berada dalam lingkup database oracle ke dalam lingkup big data dengan menggunakan DBeaver. Tabel *read* merupakan tabel yang bersumber dari tabel oracle. Kolom *INSERTED_DT* berfungsi untuk memberikan informasi tanggal berdasarkan partisi offload data tersebut dilakukan. Proses yang terakhir dilakukan setelah diproses yaitu melakukan *write* ke dalam tabel big data *RAKC_DLY_VALAS_TRANS_FACT*.

B. Testing

Tahap *testing* merupakan tahap percobaan terhadap *mapping* yang telah selesai melalui tahap *development*. Pada tahap ini, testing dilakukan dengan menggunakan aplikasi Informatica PowerCenter Workflow. Seluruh *mapping* yang telah selesai dibuat akan dimasukkan ke dalam *workflow* agar menjadi satu alur kerja yang dapat dijalankan secara otomatis dengan aplikasi Informatica PowerCenter Workflow. Tujuan dari tahap *testing* yaitu untuk memastikan bahwa *mapping* sudah berjalan sesuai dengan requirement dan data yang dihasilkan ke dalam tabel sudah sesuai dengan format serta keinginan dari user.



Gambar 3. 27 Workflow RAKC

Gambar 3.27 merupakan workflow yang dibentuk dalam proyek RAKC. Workflow tersebut dibentuk agar mapping dan proses data dapat dijalankan secara otomatis setelah urutannya telah ditentukan oleh developer. Dalam workflow RAKC, mapping dijalankan secara paralel dengan urutan dari TEMP_1, TEMP_2 dan FACT (Target akhir). Workflow tersebut kemudian akan didaftarkan melalui CTM agar menjadi satu job yang memiliki jadwal untuk dijalankan secara otomatis.

3.2.2 Proyek Plafond Efektif *Monthly*

A. *Development*

Proyek Plafond Efektif *Monthly* merupakan proyek yang berhubungan dengan nilai transaksi plafond yang diturunkan per bulan. Pada proyek ini, kegiatan yang dilakukan yaitu menambahkan kolom baru yang berisikan data transaksi dari rata-rata plafond efektif *monthly*. Penambahan kolom tersebut dilakukan dengan menambahkan kolom pada tabel dan mapping yang sudah

ada sebelumnya menggunakan tools Informatica PowerCenter Designer. Ketika perubahan kolom dilakukan pada Informatica PowerCenter Designer, maka tabel yang menjadi target data tersebut akan dimasukkan juga harus diganti, apabila tidak diganti maka tabel akan error karena menerima data dengan kolom yang dianggap tidak ada di tabel database Oracle. Penyesuaian penambahan kolom di Oracle harus ditambahkan secara manual menggunakan query SQL yang di execute di Oracle.

```
ALTER TABLE BCALS.TEMP_ALL_COMM_MTHLY_BAL_FACT  
ADD (PLAFOND_EFF_GBK_IDR NUMBER,  
PLAFOND_EFF_GBK_OCUR NUMBER);
```

Gambar 3. 28 Query Changes ALL COMM

Gambar 3.28 merupakan query yang digunakan untuk menambahkan kolom baru terhadap tabel yang telah mengalami perubahan pada mapping di Informatica PowerCenter Designer. Query yang digunakan yaitu ALTER dengan fungsi ADD dan untuk menambahkan kolom baru sesuai dengan tipe data yang telah ditentukan di dalam query tersebut.

B. Testing

Proses testing yang dilakukan dalam proyek Plafond Efektif Monthly menggunakan aplikasi Informatica PowerCenter Workflow. Seluruh mapping yang terdampak akan dilakukan *refresh mapping* dan akan dijalankan untuk memasukkan data yang baru ke dalam kolom yang baru ditambahkan. Status testing akan dimonitor melalui aplikasi Informatica PowerCenter Monitoring agar seluruh status seperti jumlah data yang masuk, kecepatan data masuk dan tingkat efektif mapping dalam memproses data ke dalam tabel dapat diawasi. Monitoring tersebut juga berfungsi untuk melihat error log ketika muncul error, sehingga error dapat diatasi dengan mudah.

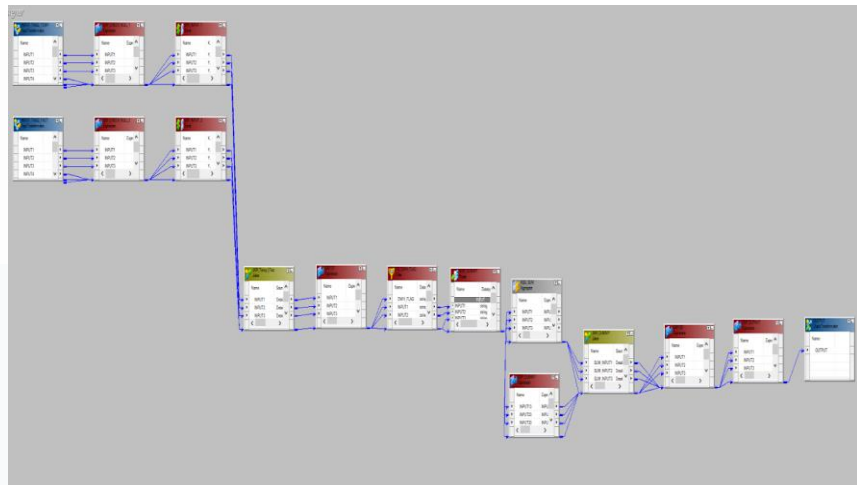
Proses testing tersebut harus dilakukan agar dapat dipastikan bahwa data yang ditujukan untuk masuk ke dalam kolom yang baru telah masuk ke kolom yang benar.

3.2.3 Proyek *Deleted Rowkey* HBASE

Proyek Deleted Rowkey HBASE merupakan proyek kerjasama antara tim DWH dan tim HBASE. Data yang diturunkan oleh tim DWH akan diproses ke dalam HBASE, tetapi HBASE memiliki struktur penyimpanan data yang berbeda dari DWH. DWH menyimpan data dalam tabel yang terdiri dari baris dan kolom, sedangkan HBASE tidak terdiri dari baris dan kolom melainkan terdiri dari *key* dan *value*. HBASE menggunakan *key* sebagai *key* unik yang mewakili *value* dari suatu data. Dalam proyek ini, rowkey digunakan sebagai *key* di dalam penyimpanan data HBASE. Proyek ini memiliki fungsi untuk melaraskan data dari DWH dan HBASE sehingga apabila ada data yang terhapus di DWH akan dapat dihapus di HBASE juga. Data yang dihapus hanya diketahui ketika data dari tim aplikasi di proses ke dalam tabel DWH, oleh karena itu DWH harus menurunkan rowkey untuk diberikan kepada tim HBASE agar dapat menyamakan perubahan data setiap harinya. Rowkey memiliki format yang berupa *value* dari beberapa kolom yang memiliki *value* unik.

A. *Development*

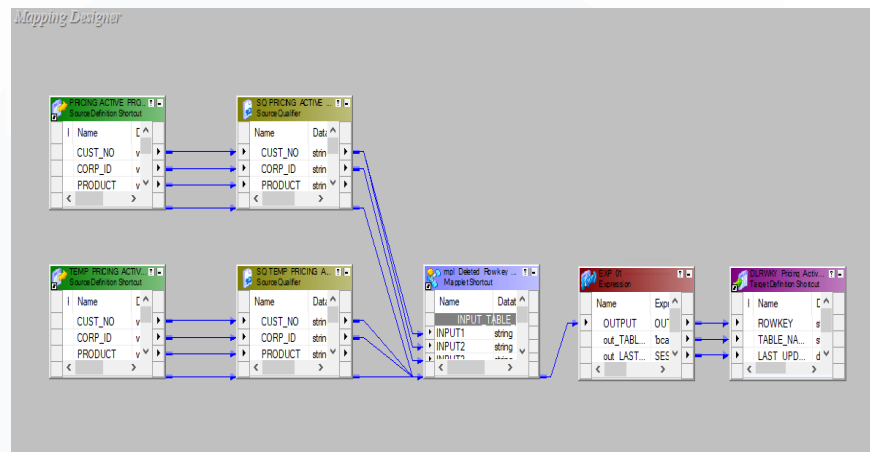
Tahap *development* yang dilakukan untuk menghasilkan rowkey dari data-data yang terhapus yaitu dengan membuat *dynamic mapplet*. Dynamic mapplet merupakan mapping yang dapat digunakan berkali-kali dalam mapping yang berbeda, hal ini dilakukan karena penghapusan data tidak hanya dilakukan dalam satu mapping dan satu tabel saja melainkan dapat dilakukan dalam banyak mapping dan tabel.



Gambar 3. 29 Dynamic Mapplet Deleted Rowkey

Gambar 3.29 merupakan *dynamic mapplet* yang digunakan untuk menghasilkan rowkey terhadap data-data yang telah terhapus. Cara kerja *dynamic mapplet* diatas yaitu dengan membandingkan data yang turun pada hari ini dengan data yang turun pada hari sebelumnya. Untuk mendapatkan data yang turun pada hari ini, data diambil dari tabel temporary yang menyimpan proses data terbaru sebelum kemudian dimasukkan ke dalam tabel yang akhir, sedangkan untuk mendapatkan data yang turun pada hari sebelumnya, data diambil dari tabel akhir karena sebelum tabel temporary melakukan proses ke tabel akhir, tabel akhir tersebut masih memiliki data dengan posisi hari sebelumnya. Setelah kedua tabel sumber telah dimasukkan, maka akan dilakukan pengecekan nilai null atau kosong sebelum data tersebut di *join*. Data kemudian di *join* untuk melakukan perbandingan apabila ada data yang turun pada tabel temporary tidak ada di dalam tabel akhir dengan posisi data hari sebelumnya, maka akan ditandai dengan *flag* 'D' yang memiliki arti bahwa data tersebut terhapus. Selanjutnya proses filter dilakukan untuk mengambil data yang hanya memiliki *flag* 'D' saja, karena *rowkey* yang ingin dihasilkan hanyalah berasal dari data yang

terhapus. Ketika proses pemisahan data selesai dilakukan, format data yang terhapus akan diubah menjadi format rowkey.



Gambar 3. 30 Contoh penggunaan dynamic maplet

Gambar 3.30 merupakan contoh penerapan *dynamic maplet* untuk menghasilkan rowkey dari data suatu tabel yang telah dihapus. Data *temporary* yang berisikan data terbaru dibandingkan dengan tabel akhir yang memiliki isi data pada hari sebelumnya, kemudian proses penghasilan rowkey terhadap data yang dihapus terjadi di dalam *dynamic maplet* yang ditandai dengan warna biru muda. Proses terakhir yaitu menuliskan output yang dihasilkan dari *dynamic maplet* ke dalam file yang akan digunakan oleh tim HBASE untuk menyamakan penghapusan data yang terjadi di DWH.

File yang telah dihasilkan setelah melalui proses *dynamic maplet* tersebut kemudian akan digabungkan menjadi satu atau melalui proses *merge*. Proses *merge* dilakukan ketika rowkey suatu proyek dihasilkan dari beberapa tabel yang berbeda sehingga menghasilkan banyak file, oleh karena itu harus dilakukan proses *merge*. Proses *merge* dilakukan dengan menggunakan file `merge_deleted_rowkey.sh` yang akan melakukan proses *merge* secara otomatis.

```

merge_deleted_rowkey.sh
1 #####
2 #
3 # Script merge file .txt deleted rowkey Hbase
4 # Modified by : Hansen Thadeus, JEL
5 #
6 #####
7
8 # path_file : path for source file .txt
9 # output_file_name : name for output file
10 # path_backup : path for backup source file .txt
11
12 path_file=$1
13 output_file_name=$2
14 path_backup=$3
15 tgl=`date +%Y_%m_%d`
16
17 count_file=$(find $path_file -type f -name "*.txt" | wc -l)
18 output_path_file=/datawh/daily/output/deleted_rowkey/
19
20 if [ -f $output_path_file$output_file_name ]; then
21 rm $output_path_file$output_file_name
22 fi
23
24 while [ $count_file -gt 0 ]
25 do
26
27 first_file=$(ls $path_file | head -n 1)
28
29 cd $path_file
30
31 #if [[ ! "$first_file" = "Deleted_Rowkey_*" ]]; then
32
33 cat $first_file >> $output_path_file$output_file_name
34 echo "Path output file : $output_path_file$output_file_name"
35 #echo >> $output_path_file$output_file_name
36 chmod 777 $first_file
37 mv $first_file $path_backup$tgl$first_file
38
39 #fi
40
41 count_file=$((count_file - 1))
42 done
43 echo "Folder is empty"
44
45 #zip file
46 backup_filename="Backup_Deleted_Rowkey"
47
48 if [ -r $path_backup$tgl$first_file.zip ]
49 then
50
51 zip -j -g $path_backup$tgl$backup_filename.zip "$path_backup$tgl"*.txt
52 rm "$path_backup$tgl"*.txt
53
54 else
55
56 zip -j $path_backup$tgl$backup_filename.zip "$path_backup$tgl"*.txt
57 rm "$path_backup$tgl"*.txt
58
59 fi

```

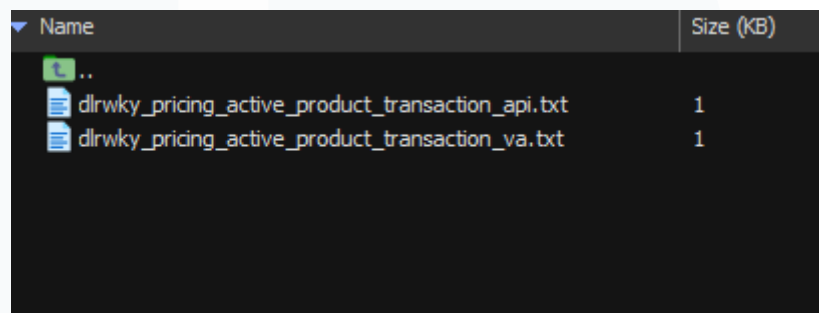
Gambar 3. 31 merge_deleted_rowkey.sh

Gambar 3.31 merupakan isi dari file merge_deleted_rowkey.sh yaitu berisikan kumpulan script dengan menggunakan bahasa pemrograman unix. Fungsi dari script Script tersebut yaitu untuk menggabungkan banyak file menjadi satu dan melakukan backup terhadap file tersebut, script ini memiliki 3 parameter yang harus diisi ketika ingin digunakan yaitu path_file, output_file_name dan path_backup. Proses yang dilakukan pertama yaitu melakukan pengecekan untuk memastikan bahwa file yang ingin digabungkan ada dalam lokasi folder yang telah ditentukan, kemudian file yang telah digabungkan akan dipindahkan ke folder backup untuk di zip menjadi satu.

B. Testing

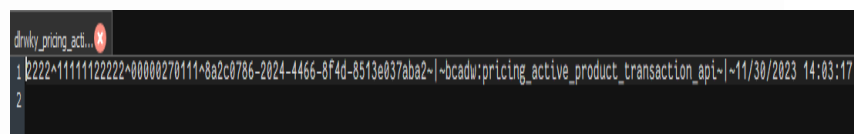
Tahap testing yang dilakukan yaitu dengan menjalankan mapping yang menggunakan dynamic mapplet untuk melakukan

proses pengambilan rowkey yang terhapus. Tahap testing dilakukan dengan menggunakan Informatica PowerCenter Workflow dengan alur kerja di dalamnya. Informatica PowerCenter Monitoring digunakan untuk memantau proses dari mapping *deleted* rowkey tersebut. Ketika proses sudah berhasil berjalan, maka proses tersebut akan menghasilkan file dengan format rowkey didalamnya.



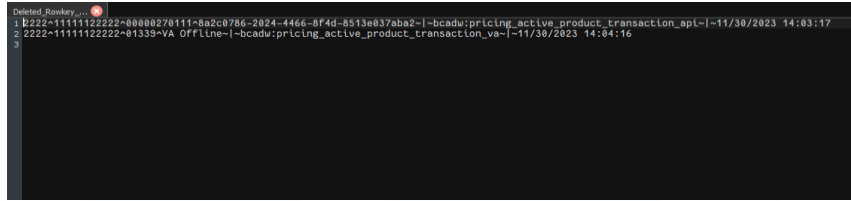
Gambar 3. 32 Output Mapping Deleted Rowkey

Gambar 3.32 merupakan output yang dihasilkan ketika menjalankan mapping *deleted rowkey*. File dengan format .txt dan prefix “dlrwky” di depan setiap file akan terbentuk secara otomatis dengan isi rowkey yang terhapus.



Gambar 3. 33 Isi dari Output Deleted Rowkey

Gambar 3.32 merupakan hasil dari salah satu file yang di proses. Data tersebut merupakan format dari rowkey data yang terhapus.



```
Deleted_Rowkey_...
1 2222~11111122222~8888879111~832c8786~3824~4466~8f4d~8513e837aba2~|~bcadu:pricing_active_product_transaction_apl~|~11/30/2023 14:03:17
2 2222~11111122222~91339~VA Offline~|~bcadu:pricing_active_product_transaction_va~|~11/30/2023 14:04:16
3
```

Gambar 3. 34 Output merge_deleted_rowkey.sh

Gambar 3.34 merupakan hasil ketika file deleted rowkey yang dihasilkan di merge. Dapat dilihat bahwa dalam file tersebut terdapat dua baris data yang berasal dari sumber tabel berbeda. File deleted rowkey yang sudah digabung akan digunakan oleh tim HBASE untuk menghapus data-data tersebut di dalam lingkup HBASE sehingga data dari tim DWH dan HBASE tidak mengalami perbedaan.

3.2.4 Proyek Avg 12 Mth Pricing

A. Development

Proyek Avg 12 Mth Pricing merupakan proyek penambahan kolom terhadap tabel yang digunakan untuk proyek Pricing. Penambahan kolom dilakukan menggunakan mapplet yang memiliki fungsi untuk mengambil data transaksi selama 12 bulan dan di rata-rata. Penambahan kolom tersebut dilakukan dengan menambahkan kolom pada tabel dan mapping yang sudah ada sebelumnya menggunakan tools Informatica PowerCenter Designer. Ketika perubahan kolom dilakukan pada Informatica PowerCenter Designer, maka tabel yang menjadi target data tersebut akan dimasukkan juga harus diganti, apabila tidak diganti maka tabel akan error karena menerima data dengan kolom yang dianggap tidak ada di tabel database Oracle. Penyesuaian penambahan kolom di Oracle harus ditambahkan secara manual menggunakan query SQL yang di execute di Oracle.

```

ALTER TABLE BCALS.TEMP_COMM_MTHLY_BAL_FACT_1
ADD (AVG_12MTH_COMM_CURR_AMT_OCUR NUMBER,
AVG_12MTH_COMM_CURR_AMT_IDR NUMBER,
AVG_12MTH_COMM_USED_AMT_OCUR NUMBER,
AVG_12MTH_COMM_USED_AMT_IDR NUMBER);

ALTER TABLE BCALS.TEMP_COMM_MTHLY_BAL_FACT_2
ADD (AVG_12MTH_COMM_CURR_AMT_OCUR NUMBER,
AVG_12MTH_COMM_CURR_AMT_IDR NUMBER,
AVG_12MTH_COMM_USED_AMT_OCUR NUMBER,
AVG_12MTH_COMM_USED_AMT_IDR NUMBER);

ALTER TABLE BCALS.TEMP_COMM_MTHLY_BAL_FACT_3
ADD (AVG_12MTH_COMM_CURR_AMT_OCUR NUMBER,
AVG_12MTH_COMM_CURR_AMT_IDR NUMBER,
AVG_12MTH_COMM_USED_AMT_OCUR NUMBER,
AVG_12MTH_COMM_USED_AMT_IDR NUMBER);

```

Gambar 3. 35 Query Proyek AVG 12 MTH Pricing

Gambar 3.35 merupakan query yang digunakan untuk menambahkan kolom baru terhadap tabel yang telah mengalami perubahan pada mapping di Informatica PowerCenter Designer. Query yang digunakan yaitu ALTER dengan fungsi ADD dan untuk menambahkan kolom baru sesuai dengan kolom-kolom baru yang ditambahkan ke dalam mapping.

B. Testing

Proses testing yang dilakukan dalam proyek Proyek Avg 12 Mth Pricing menggunakan aplikasi Informatica PowerCenter Workflow. Seluruh mapping yang terdampak akan dilakukan *refresh mapping* dan akan dijalankan untuk memasukkan data yang baru ke dalam kolom yang baru ditambahkan. Status testing akan dimonitor melalui aplikasi Informatica PowerCenter Monitoring agar seluruh status seperti jumlah data yang masuk, kecepatan data masuk dan tingkat efektif mapping dalam memproses data ke dalam tabel dapat diawasi. Monitoring tersebut juga berfungsi untuk melihat error log

ketika muncul error, sehingga error dapat diatasi dengan mudah. Proses testing tersebut harus dilakukan agar dapat dipastikan bahwa data yang ditujukan untuk masuk ke dalam kolom yang baru telah masuk ke kolom yang benar.

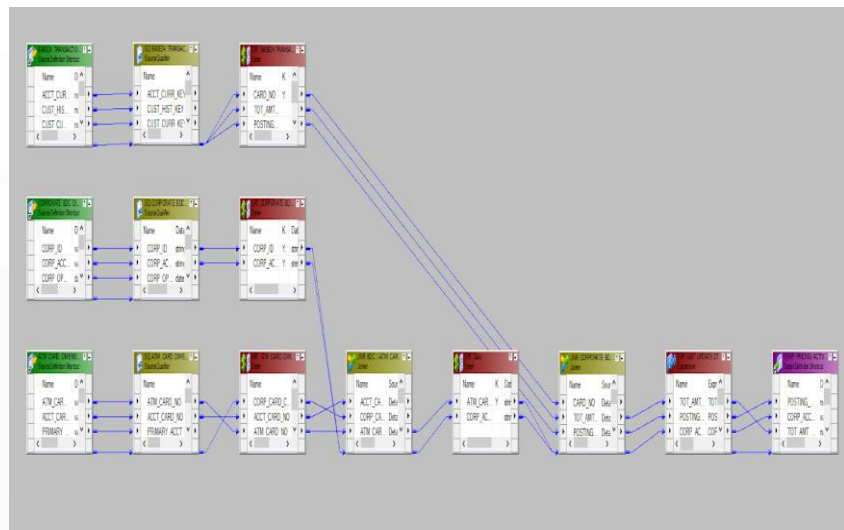
3.2.5 Proyek Pricing *Active & History Product BDC*

Proyek Pricing merupakan proyek yang di request oleh tim aplikasi pricing dengan tujuan untuk membantu pengajuan penurunan suku bunga. Data-data yang diproses di DWH akan dikirimkan ke dalam aplikasi Pricing yang digunakan oleh cabang. Proyek pricing *Active & History Product BDC* digunakan untuk segmen *business debit card (BDC)* yang terdiri dari 3 jenis kartu yaitu Petty Cash, Loyalty dan Deposit. Kartu-kartu tersebut memiliki masa aktif kartu, oleh karena itu data Pricing BDC dibagi menjadi dua yaitu *Active* dan *History*. Data Pricing BDC *Active* berisikan dengan data transaksi kartu yang masih aktif dengan artian kartu masih aktif digunakan untuk melakukan transaksi dan belum kadaluarsa, sedangkan Pricing BDC *History* berisikan dengan data transaksi kartu yang sudah tidak aktif atau kadaluarsa.

A. *Development*

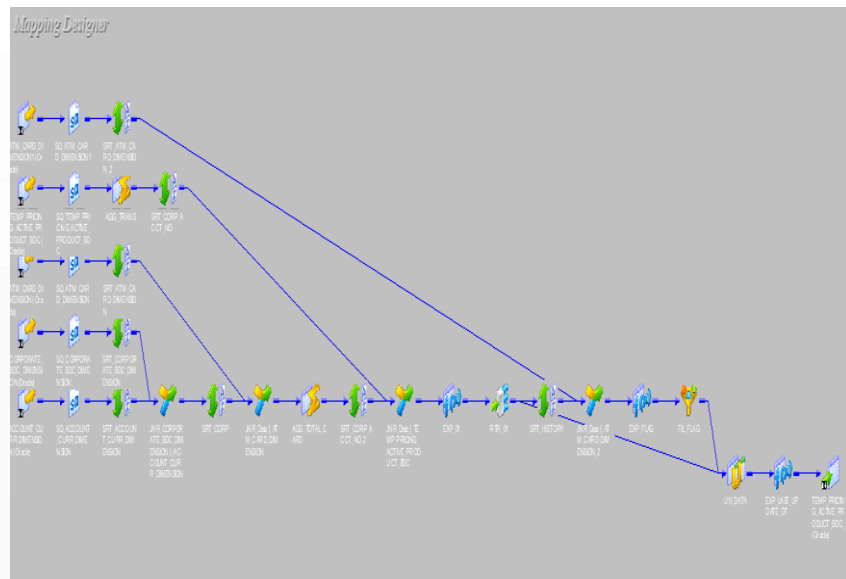
Tahap development yang dilakukan untuk proyek Pricing *Active & History Product BDC* dipisah menjadi dua tahap yaitu tabel temporary dan tabel akhir pricing bdc.

- **Pricing *Active Product BDC***



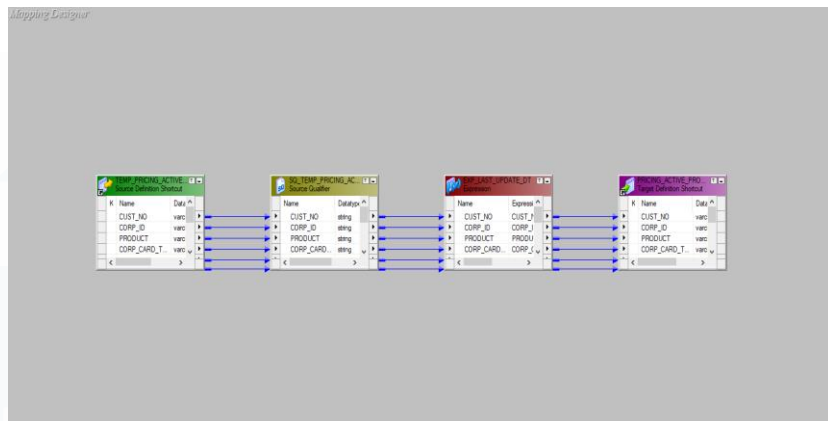
Gambar 3. 36 Mapping Temporary 1 Pricing Active Product BDC

Gambar 3.36 merupakan mapping tabel temporary dari proyek Pricing Active Product BDC. Dalam mapping tersebut, data yang diambil yaitu data transaksi setiap kartu selama satu tahun kebelakang dari tanggal hari ini. Pengambilan data transaksi ini dipisah dari mapping utama karena proses pengambilan data transaksi dari tabel base24 akan memakan waktu yang cukup lama karena jumlah data transaksi yang besar. Tabel base24 tersebut di join dengan tabel ATM dan BDC sehingga data transaksi yang diambil hanyalah data transaksi kartu yang dimiliki oleh korporat BDC.



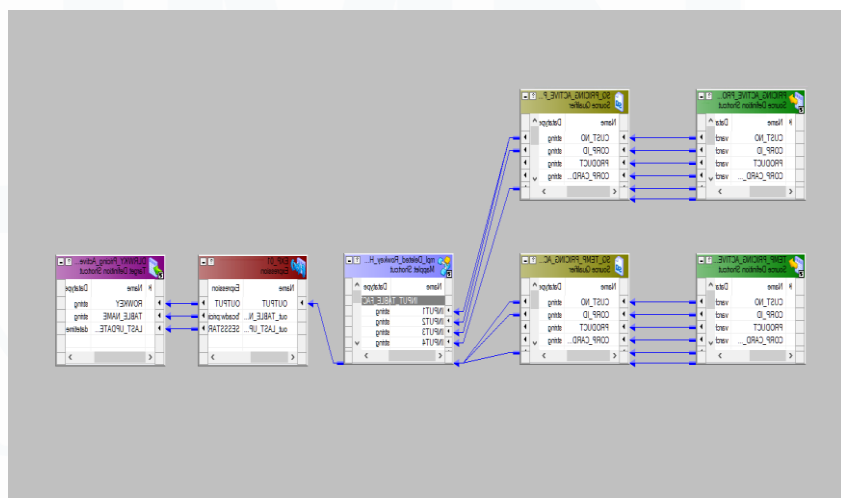
Gambar 3. 37 Mapping Temporary 2 Pricing Active Product BDC

Gambar 3.37 merupakan mapping temporary Pricing Active Product BDC yang ke dua, dalam mapping tersebut terdapat beberapa sumber data yang harus digabungkan menjadi satu karena ada kolom yang dibutuhkan untuk data dalam tabel Pricing Active Product BDC. Terdapat beberapa proses transformasi yang dilakukan untuk mendapatkan data Pricing Active Product BDC, salah satunya yaitu *aggregator* untuk menghitung rata-rata total transaksi dan frekuensi transaksi dari setiap kartu. Setelah mendapatkan semua data yang diperlukan, validasi dilakukan dalam mapping tersebut dengan menggunakan transformasi *router* untuk memisahkan data yang aktif dan history. Validasi ini dilakukan karena terdapat rekening yang aktif tetapi tidak memiliki kartu dan terdapat rekening yang masih aktif tetapi kartu yang dimiliki telah kadaluarsa, sehingga dengan adanya validasi tersebut data yang diambil hanya rekening yang aktif tetapi tidak memiliki kartu sedangkan yang lainnya akan masuk ke dalam history.



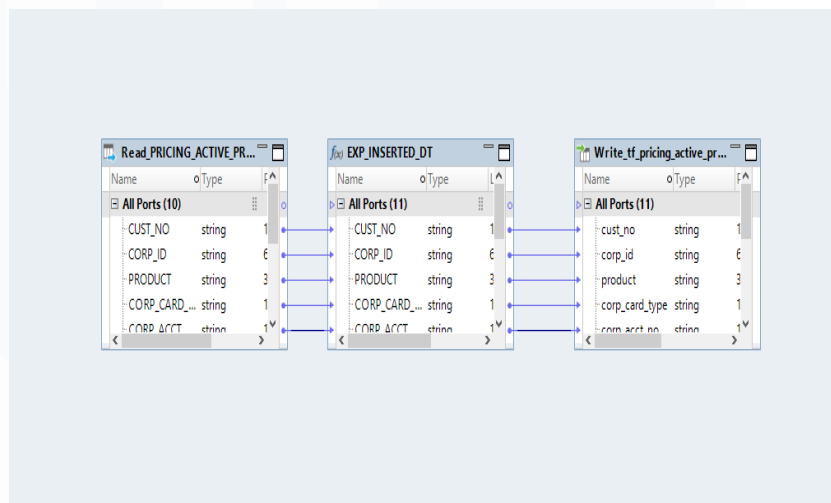
Gambar 3. 38 Mapping Fact Pricing Active Product BDC

Gambar 3.38 merupakan proses terakhir dari pembentukan tabel Pricing Active Product BDC. Proses yang dilakukan yaitu proses load dengan memasukkan data yang telah melalui proses *extract* dan *transform*. Sebelum data dimasukkan ke dalam target tabel akhir yaitu Pricing Active Product BDC, dilakukan penambahan transformasi *expression* untuk menambahkan kolom *LAST_UPDATE_DT* yang berfungsi untuk mengeluarkan tanggal dan waktu secara otomatis pada saat data diproses. Kemudian langkah yang terakhir yaitu melakukan *push* data ke dalam tabel target akhir di database Oracle.



Gambar 3. 39 Mapping Deleted Rowkey Pricing Active Product BDC

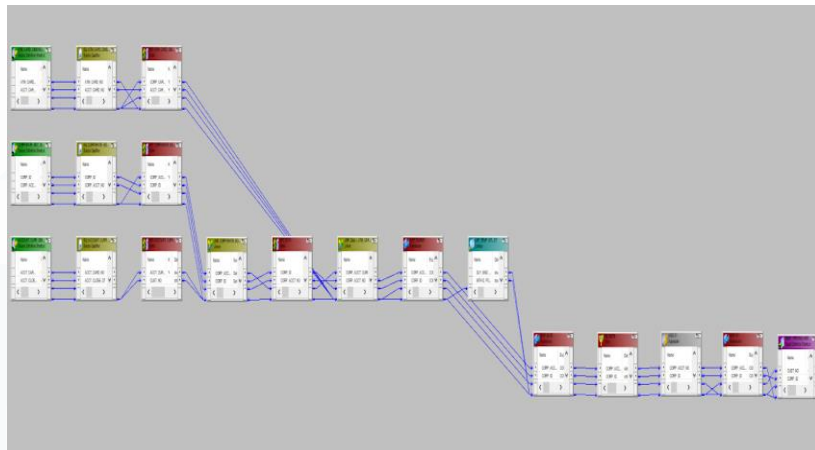
Gambar 3.39 merupakan proses mapping untuk menghasilkan rowkey daripada setiap data yang terhapus karena data dari Pricing Active BDC bersifat dapat di update sehingga akan ada pergerakan data setiap harinya. Output yang dihasilkan dari mapping deleted rowkey Pricing Active BDC berbentuk file dengan format .txt yang akan diberikan kepada tim HBASE agar data yang disimpan dalam DWH dan HBASE selaras.



Gambar 3. 40 Mapping Offload Pricing Active Product BDC

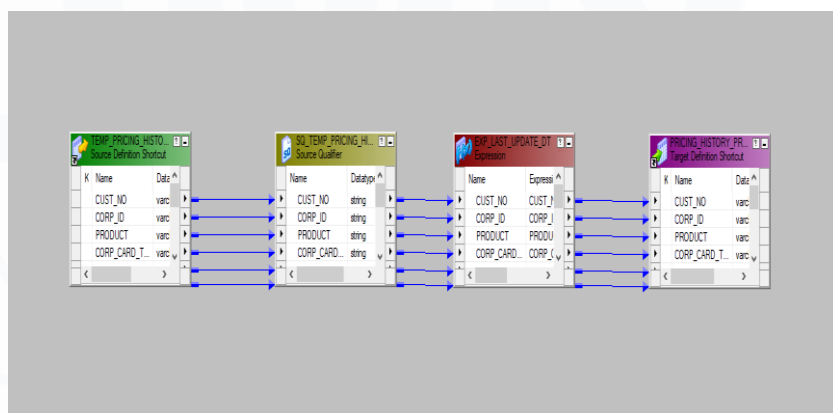
Gambar 3.40 merupakan mapping yang melakukan proses offload data Pricing Active Product BDC. Proses offload dilakukan dengan menggunakan aplikasi Informatica Developer. Proses offload ini berfungsi untuk memasukkan data yang berada dalam lingkup database oracle ke dalam lingkup big data dengan menggunakan DBeaver. Tabel *read* merupakan tabel yang bersumber dari tabel oracle. Kolom INSERTED_DT berfungsi untuk memberikan informasi tanggal berdasarkan partisi offload data tersebut dilakukan. Proses yang terakhir dilakukan setelah diproses yaitu melakukan *write* ke dalam tabel big data Pricing Active Product BDC.

- **Pricing History Product BDC**



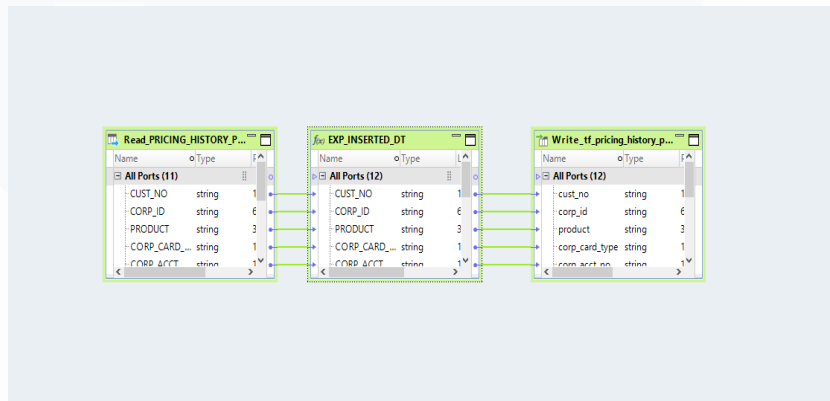
Gambar 3. 41 Mapping Temporary 1 Pricing History Product BDC

Gambar 3.41 merupakan mapping *temporary* dari data mart pricing history product BDC. Dalam mapping tersebut, data pricing history BDC diambil dari beberapa tabel yang memiliki informasi kartu korporat BDC yang telah kadaluarsa atau tidak aktif lagi. Periode data yang diambil untuk kartu korporat BDC yang telah kadaluarsa yaitu selama satu tahun kebelakang dari hari ini. Kolom-kolom yang sudah diambil dari beberapa tabel yang dibutuhkan kemudian akan digabungkan dengan menggunakan transformasi *joiner*. Kemudian data yang sudah digabungkan akan di filter untuk mengambil data selama satu tahun terakhir saja.



Gambar 3. 42 Mapping Fact Pricing History Product BDC

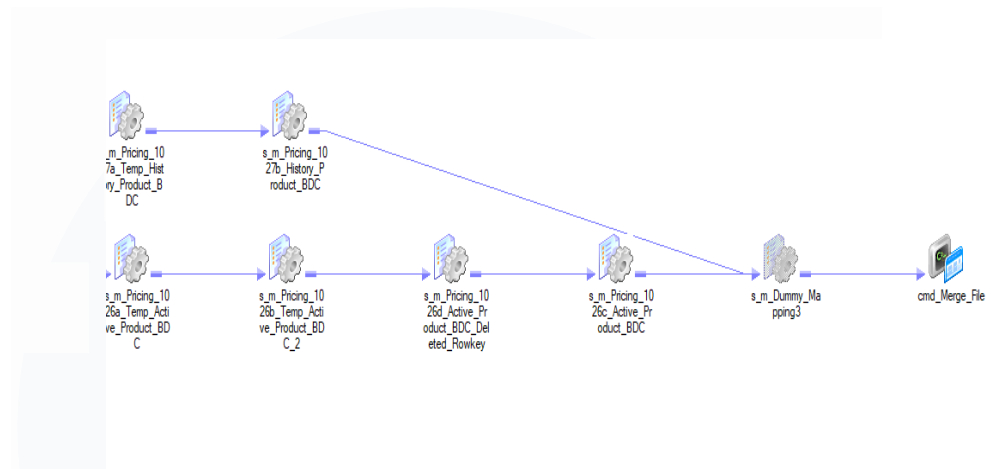
Gambar 3.42 merupakan proses terakhir dari pembentukan tabel Pricing History Product BDC. Proses yang dilakukan yaitu proses load dengan memasukkan data yang telah melalui proses *extract* dan *transform*. Sebelum data dimasukkan ke dalam target tabel akhir yaitu Pricing History Product BDC, dilakukan penambahan transformasi *expression* untuk menambahkan kolom `LAST_UPDATE_DT` yang berfungsi untuk mengeluarkan tanggal dan waktu secara otomatis pada saat data diproses. Kemudian langkah yang terakhir yaitu melakukan *push* data ke dalam tabel target akhir di database Oracle.



Gambar 3. 43 Mapping Offload Pricing History Product BDC

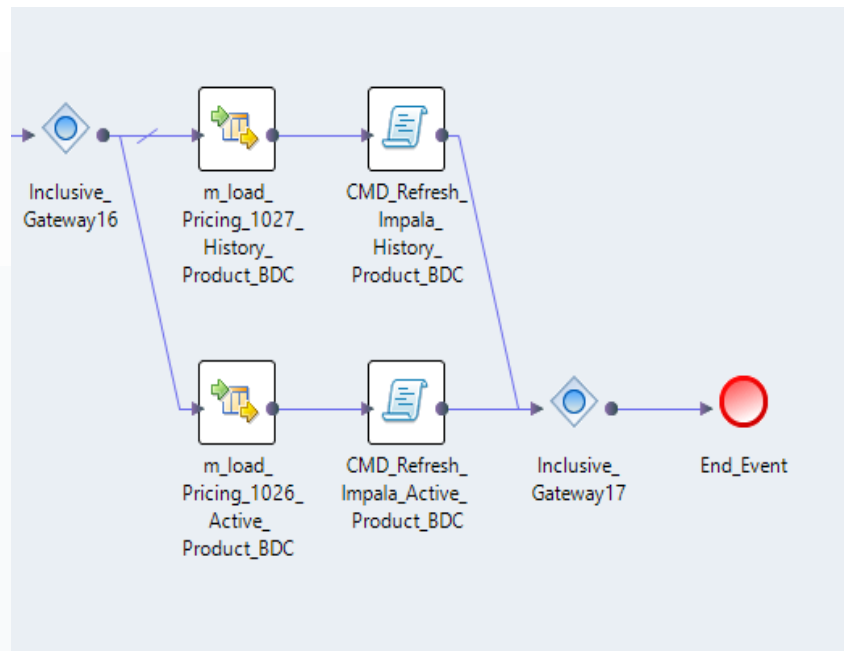
Gambar 3.43 merupakan mapping yang melakukan proses offload data Pricing History Product BDC. Proses offload dilakukan dengan menggunakan aplikasi Informatica Developer. Proses offload ini berfungsi untuk memasukkan data yang berada dalam lingkup database oracle ke dalam lingkup big data dengan menggunakan DBeaver. Tabel *read* merupakan tabel yang bersumber dari tabel oracle. Kolom `INSERTED_DT` berfungsi untuk memberikan informasi tanggal berdasarkan partisi offload data tersebut dilakukan. Proses yang terakhir dilakukan setelah diproses yaitu melakukan *write* ke dalam tabel big data Pricing History Product BDC.

B. Testing



Gambar 3. 44 Workflow Pricing BDC

Gambar 3.44 merupakan workflow yang dibentuk dalam proyek Pricing BDC. Workflow tersebut dibentuk agar mapping dan proses data dapat dijalankan secara otomatis setelah urutannya telah ditentukan oleh developer. Dalam workflow Pricing BDC. Mapping dijalankan secara paralel dengan urutan yang telah ditentukan dan dibuat dengan alur kerja. Dalam workflow Pricing BDC, terdapat satu komponen command yang berisikan parameter untuk memanggil *script* merge_deleted_rowkey.sh yang telah dibuat sebelumnya. Tujuan dari menjalankan workflow pada skema development yaitu untuk memastikan bahwa proses alur kerja dan data yang dihasilkan sudah sesuai dengan keinginan user. Workflow tersebut kemudian akan didaftarkan melalui CTM agar menjadi satu job yang memiliki jadwal untuk dijalankan secara otomatis.



Gambar 3. 45 Workflow Offload Pricing BDC

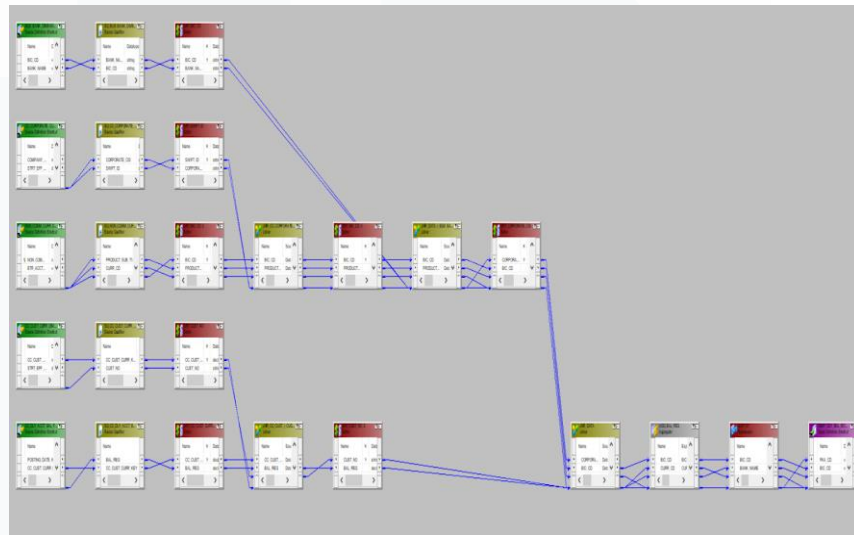
Gambar 3.45 merupakan *workflow* atau alur kerja dari mapping dan tabel offload Pricing BDC. Workflow tersebut terdiri dari inclusive gateway yang berfungsi mapping-mapping tersebut dapat berjalan secara paralel. Kemudian dalam workflow tersebut terdapat komponen command untuk melakukan refresh impala dan menentukan lokasi tabel. Workflow tersebut akan dijalankan agar dapat dipastikan bahwa mapping sudah berjalan dengan baik dan menghasilkan data *offload* yang sudah sesuai. Ketika workflow dan data sudah sesuai maka workflow tersebut akan di *deploy* ke dalam application sehingga dapat berjalan secara otomatis dengan *scheduler*.

3.2.6 Proyek Belief

A. *Development*

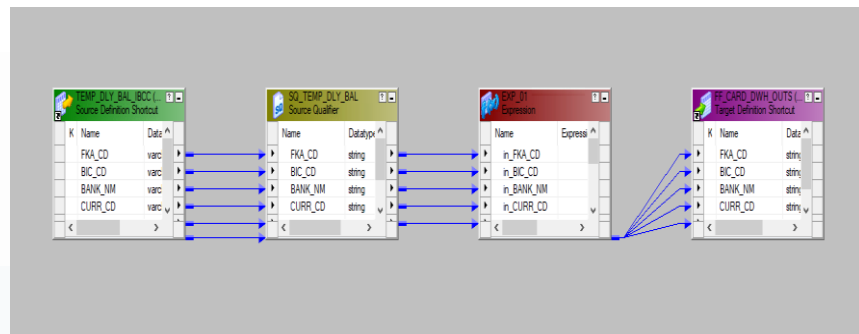
Proyek belief merupakan proyek yang diajukan oleh tim aplikasi belief dengan tujuan untuk memasukkan data produk IBCC ke dalam aplikasi yang digunakan oleh tim belief. Produk IBCC adalah

produk visa corporate credit card bank. Proses development dalam proyek ini terbagi menjadi dua tahap yaitu pembuatan tabel temporary dan proses penghasilan output data yang dibutuhkan.



Gambar 3. 46 Mapping Temporary Belief

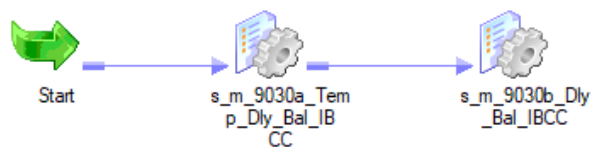
Gambar 3.46 merupakan mapping dari pembuatan tabel temporary. Mapping tersebut dibentuk dengan tujuan untuk mengambil data yang diperlukan oleh tim aplikasi belief untuk produk visa corporate credit card bank. Proses yang terjadi dalam mapping temporary yaitu pemilihan kolom yang dibutuhkan dari sumber data yang telah ditentukan dan ketika semua kolom sudah dipilih, maka seluruh kolom tersebut akan digabungkan menjadi satu menggunakan transformasi *joiner*, kolom-kolom tersebut digabungkan berdasarkan kondisi yang telah ditentukan. Kemudian mapping tersebut ditambahkan kolom `LAST_UPDATE_DT` untuk memberikan informasi waktu ketika mapping tersebut dijalankan.



Gambar 3. 47 Mapping Akhir Belief

Gambar 3.47 merupakan proses terakhir dari pembentukan mapping proyek belief. Proses yang dilakukan yaitu proses load dengan memasukkan data yang telah melalui proses *extract* dan *transform*. Dalam proses terakhir proyek belief, terdapat komponen *expression* dengan nama EXP_01 yang memiliki fungsi untuk membentuk format penulisan data ke dalam file. Berbeda dengan mapping output yang lain, tim aplikasi dari belief melakukan request untuk menurunkan data tersebut ke dalam bentuk file dengan format .txt dan menggunakan delimiter '~|~'.

B. Testing



Gambar 3. 48 Workflow Proyek Belief

Gambar 3.48 merupakan workflow yang dibentuk dalam proyek belief. Workflow tersebut dibentuk agar mapping dan proses

data dapat dijalankan secara otomatis setelah urutannya telah ditentukan oleh developer. Dalam workflow belief, mapping dijalankan secara paralel dengan urutan dari mapping temporary dan mapping akhir (Target akhir). Tujuan dari menjalankan workflow pada skema development yaitu untuk memastikan bahwa proses alur kerja dan data yang dihasilkan sudah sesuai dengan keinginan user. Workflow tersebut kemudian akan didaftarkan melalui CTM agar menjadi satu job yang memiliki jadwal untuk dijalankan secara otomatis.

3.2.7 Proyek NPWP 16 Digit

A. *Development*

Proyek NPWP 16 Digit merupakan proyek penambahan length terhadap kolom yang digunakan untuk data NPWP. Penambahan panjang kolom dilakukan terhadap mapping-mapping dari sumber data, seluruh mapping yang bersangkutan dan mapping akhir yang memiliki fungsi untuk menghasilkan kolom data NPWP. Penambahan panjang kolom tersebut dilakukan dengan menambahkan panjang kolom pada tabel dan mapping yang sudah ada sebelumnya menggunakan tools Informatica PowerCenter Designer. Ketika perubahan panjang kolom dilakukan pada Informatica PowerCenter Designer, maka tabel yang menjadi target data tersebut akan dimasukkan juga harus diganti, apabila tidak diganti maka tabel akan error karena menerima data dengan panjang kolom yang terlalu besar di tabel database Oracle. Penyesuaian penambahan panjang kolom di Oracle harus ditambahkan secara manual menggunakan query SQL yang di execute di Oracle.

```

--ACCT_LOAN_CURR_DIMENSION
ALTER TABLE BCALS.TEMP_ACCT_LOAN_CURR_DIM_1
MODIFY TAX_NO VARCHAR2(16);

ALTER TABLE BCALS.TEMP_ACCT_LOAN_CURR_DIM_2
MODIFY TAX_NO VARCHAR2(16);

ALTER TABLE BCALS.ACCT_LOAN_CURR_DIMENSION
MODIFY TAX_NO VARCHAR2(16);

--FF_BD_ACCT_LOAN_CURR_DIMENSION diganti juga jadi 16

--ACCT_LOAN_HIST_DIMENSION
ALTER TABLE BCALS.ACCT_LOAN_HIST_DIMENSION
MODIFY TAX_NO VARCHAR2(16);

--COLL_CURR_DIMENSION
ALTER TABLE BCALS.TEMP_COLL_CURR_DIMENSION_1
MODIFY COLL_NWP_PEMILIK_JAMINAN VARCHAR2(16);

ALTER TABLE BCALS.TEMP_COLL_CURR_DIMENSION_2
MODIFY COLL_NWP_PEMILIK_JAMINAN VARCHAR2(16);

ALTER TABLE BCALS.COLL_CURR_DIMENSION
MODIFY COLL_NWP_PEMILIK_JAMINAN VARCHAR2(16);

```

Gambar 3. 49 Query Proyek NPWP 16 Digit

Gambar 3.49 merupakan query yang digunakan untuk menambahkan panjang kolom terhadap tabel yang telah mengalami perubahan pada mapping di Informatica PowerCenter Designer. Query yang digunakan yaitu ALTER dengan fungsi MODIFY untuk menambahkan panjang kolom tertentu sesuai dengan kolom-kolom yang mengalami perubahan panjang kolom.

B. Testing

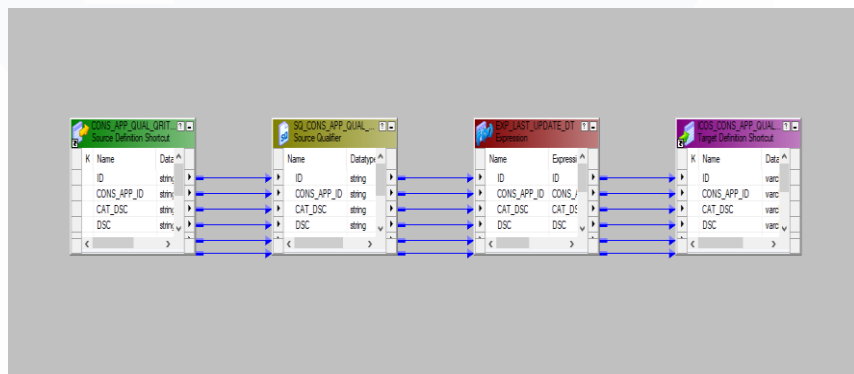
Proses testing yang dilakukan dalam proyek Proyek NPWP 16 Digit menggunakan aplikasi Informatica PowerCenter Workflow. Seluruh mapping yang terdampak akan dilakukan *refresh mapping* dan akan dijalankan untuk memasukkan data yang baru ke dalam kolom yang baru dipanjangkan. Status testing akan dimonitor melalui aplikasi Informatica PowerCenter Monitoring agar seluruh status seperti jumlah data yang masuk, kecepatan data masuk dan tingkat efektif mapping dalam memproses data ke dalam tabel dapat diawasi. Monitoring tersebut juga berfungsi untuk melihat error log

ketika muncul error, sehingga error dapat diatasi dengan mudah. Proses testing tersebut harus dilakukan agar dapat dipastikan bahwa data yang memiliki jumlah panjang baru masuk ke dalam kolom yang telah dipanjangkan.

3.2.8 Proyek CCOS *Qualitative Criteria*

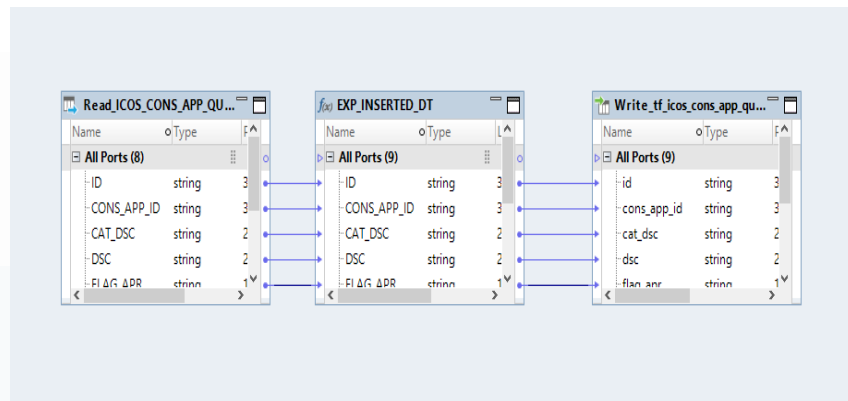
A. *Development*

Proyek CCOS *Qualitative Criteria* merupakan proyek penurunan data dari sumber data flat file yang diberikan oleh tim aplikasi CCOS ke dalam tabel Oracle. Proyek CCOS *Qualitative Criteria* akan menghasilkan satu tabel bernama ICOS_CCOS_APP_QUAL_QRIT.



Gambar 3. 50 Mapping CCOS Qualitative Criteria

Gambar 3.50 merupakan mapping dari proyek CCOS *Qualitative Criteria*. Proses yang dilakukan yaitu proses load dengan memasukkan data yang telah melalui proses *data cleansing*. Sebelum data dimasukkan ke dalam target tabel akhir yaitu ICOS_CCOS_APP_QUAL_QRIT, dilakukan penambahan transformasi *expression* untuk menambahkan kolom LAST_UPDATE_DT yang berfungsi untuk mengeluarkan tanggal dan waktu secara otomatis pada saat data diproses. Kemudian langkah yang terakhir yaitu melakukan *push* data ke dalam tabel target akhir di database Oracle.



Gambar 3. 51 Mapping Offload CCOS Qualitative Criteria

Gambar 3.51 merupakan mapping yang melakukan proses offload data proyek CCOS *Qualitative Criteria*. Proses offload dilakukan dengan menggunakan aplikasi Informatica Developer. Proses offload ini berfungsi untuk memasukkan data yang berada dalam lingkup database oracle ke dalam lingkup big data dengan menggunakan DBeaver. Tabel *read* merupakan tabel yang bersumber dari tabel oracle. Kolom *INSERTED_DT* berfungsi untuk memberikan informasi tanggal berdasarkan partisi offload data tersebut dilakukan. Proses yang terakhir dilakukan setelah diproses yaitu melakukan *write* ke dalam tabel big data CCOS *Qualitative Criteria*.

B. Testing



Gambar 3. 52 Session CCOS Qualitative Criteria

Gambar 3.52 merupakan session dari mapping proyek CCOS *Qualitative Criteria*. Session tersebut terdapat pada alur kerja atau workflow yang sudah ada sebelumnya dan digabungkan dengan mapping proyek CCOS lainnya. Proses testing yang dilakukan yaitu dengan menjalankan session tersebut secara sendiri. Tujuan dari menjalankan workflow pada skema development yaitu untuk memastikan bahwa proses alur kerja dan data yang dihasilkan sudah sesuai dengan keinginan user. Workflow tersebut kemudian akan didaftarkan melalui CTM agar menjadi satu job yang memiliki jadwal untuk dijalankan secara otomatis.

3.3 Kendala yang Ditemukan

Adapun beberapa kendala yang ditemukan selama kegiatan magang berlangsung sebagai data engineer di PT Bank Central Asia Tbk yaitu sebagai berikut:

1. Keterbatasan peserta magang dalam mengakses beberapa website internal PT Bank Central Asia Tbk seperti website untuk mengakses *requirement* dari user sehingga peserta magang kesulitan dalam mengerti tujuan proyek dan kebutuhan proyek tersebut.
2. Tidak memiliki akses untuk melakukan upload file keluar dari laptop kantor sehingga mempersulit proses pengiriman file-file kantor ataupun *screenshot* yang digunakan untuk pembuatan laporan magang.
3. Peserta magang ditempatkan di site kantor yang berbeda dari site kantor mentor sehingga komunikasi tidak dapat dilakukan secara langsung. Hal tersebut menyebabkan peserta magang tidak dapat berkomunikasi dengan mendapatkan respon yang cukup cepat dari mentor.
4. Penggunaan tools ETL yang tidak pernah diajarkan oleh kampus Universitas Multimedia Nusantara sehingga perlu melakukan pembelajaran dan beradaptasi untuk menggunakan tools ETL yang digunakan oleh PT Bank Central Asia Tbk yaitu Informatica.

3.4 Solusi atas Kendala yang Ditemukan

Solusi yang ditemukan dan digunakan untuk mengatasi seluruh kendala selama kegiatan magang berlangsung di PT Bank Central Asia Tbk sebagai *data engineer* yaitu sebagai berikut:

1. Peserta magang diberikan briefing secara terpisah untuk membahas requirement dan informasi yang peserta magang tidak dapat akses. Apabila dibutuhkan untuk melihat requirement, peserta magang akan diberikan requirement melalui screenshot yang dikirimkan melalui Microsoft Teams.
2. Peserta magang menggunakan aplikasi yang berbasis cloud untuk membuat laporan magang sehingga peserta magang dapat memasukkan screenshot mengenai proyek untuk membuat laporan magang.
3. Peserta magang diberikan izin untuk melakukan *voice call* secara langsung apabila dibutuhkan. Peserta magang juga memiliki teman satu tim di site kantor yang sama sehingga jika mentor sedang tidak bisa merespon maka dapat dibantu oleh teman satu tim yang lain.
4. Peserta magang diajarkan secara langsung oleh mentor dan teman satu tim lainnya. Kemudian peserta magang diberikan tugas dari tingkat kesulitan rendah sampai tinggi sehingga peserta magang dapat meningkatkan kemampuan dalam menggunakan tools ETL Informatica.