



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

TUGURE Technical Document

TUGURE DATA WAREHOUSE



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Document Information

Prepared By:	Husni Ridwan	Document Version Date:	23.02.2024
Reviewed By:		Preparation Date:	23.02.2024
Document Version No:	V 1.0	Review Date:	23.02.2024

Version History

Ver. No.	Ver. Date	Revised By	Description	Filename
1.0	23.02.2024	Husni Ridwan	Initial Draft	Tugure_DataWarehouse_TechnicalDocumentation 20240223 v.1.docx



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

DAFTAR ISI

DAFTAR ISI	3
KATA PENGANTAR	4
TUJUAN PEMBUATAN DOKUMEN	5
DATA WAREHOUSE	6
PENGERTIAN DAN KONSEP	6
ARSITEKTUR DIAGRAM.....	7
DATA SOURCE	7
DATA WAREHOUSE	7
BI/ Visualization.....	8
NI-FI.....	9
Pipeline 1.....	10
Pipeline 2.....	31
Pipeline 3.....	91
Pipeline 4.....	99
Pipeline 5.....	107
Pipeline 6.....	107



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

KATA PENGANTAR

KERAHASIAAN

Dokumen ini bersifat rahasia dan tidak boleh digandakan, digunakan atau diungkapkan secara keseluruhan ataupun sebagian untuk tujuan apapun selain untuk mengevaluasi isi dari dokumennya. Dengan kesepakatan, PT. Tugure Reasuransi Indonesia (Tugure) akan memiliki hak untuk menggandakan, menggunakan atau mengungkapkan data-data yang ada di dokumen ini. Hak PT. Tugure Reasuransi Indonesia (Tugure) tidak akan dibatasi untuk menggunakan informasi yang terkandung dalam data jika diperoleh dari sumber lain tanpa kerahasiaan.

HAK MILIK INTELEKTUAL

Seluruh data dan source code yang terdapat di Tugure Data Warehouse adalah hak milik. Teknologi yang akan diterapkan dalam Data Warehouse adalah hak milik PT. Trimitra Sistem Solusindo (TMS) yang dapat digunakan secara bebas dalam lingkungan teknologi Tugure.

KELALAIAN DAN MANAJEMEN PERUBAHAN

Pihak Tugure berhak untuk memperbaiki kesalahan atau kelalaian mengenai dokumen ini. Perubahan dan penambahan yang sebelumnya tidak termasuk dalam dokumen ini dapat dimasukkan dengan prosedur Permintaan Perubahan.



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

TUJUAN PEMBUATAN DOKUMEN

Tujuan dari dokumen teknis ini adalah untuk memberikan gambaran mengenai cara pemakaian dan konfigurasi secara mendetil terkait pengoperasian tools dalam environment proyek Tugure Data Warehouse. Dokumen ini juga menjelaskan hal-hal teknis dan best practice yang dianggap sering terjadi dalam proses-proses yang dilakukan tersebut.



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

DATA WAREHOUSE

PENGERTIAN DAN KONSEP

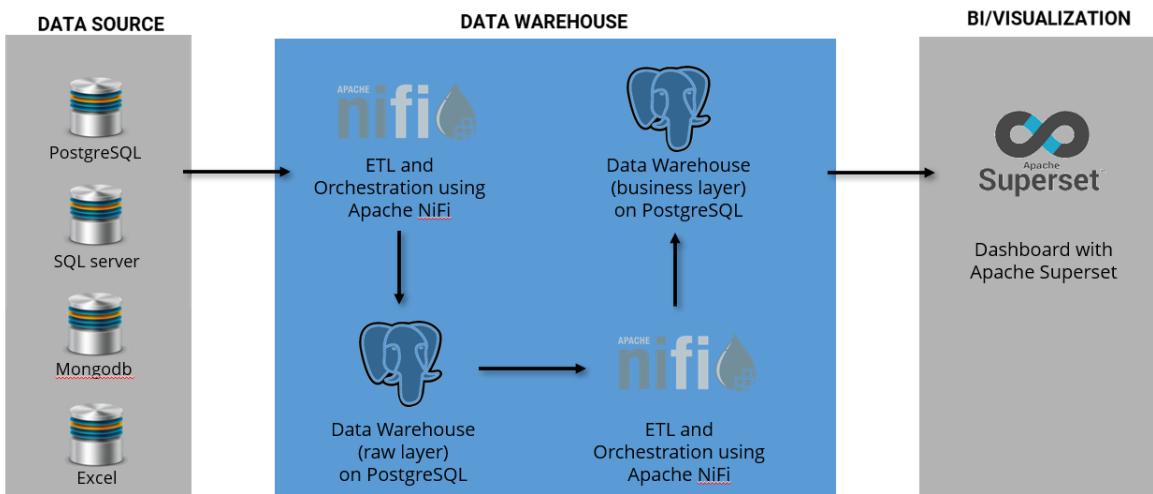
Datawarehouse adalah sebuah sistem penyimpanan data yang dirancang untuk memfasilitasi dan mendukung proses analisis data dalam suatu organisasi. Tujuan utama dari datawarehouse adalah untuk mengumpulkan data dari berbagai sumber yang tersebar, mengintegrasikannya menjadi satu lokasi penyimpanan yang terpusat, dan menyediakan akses yang mudah bagi pengguna untuk menganalisis data tersebut.

Konsep datawarehouse mencakup beberapa prinsip dan teknik yang digunakan untuk merancang, mengelola, dan menggunakan datawarehouse secara efektif. Berikut adalah beberapa konsep kunci dalam datawarehouse :

1. Integrasi Data: Datawarehouse mengintegrasikan data dari berbagai sumber yang tersebar, termasuk sistem operasional, file teks, dan data eksternal lainnya. Integrasi data memastikan bahwa data yang disimpan dalam datawarehouse konsisten dan terstandarisasi.
2. Orientasi Subjek: Datawarehouse disusun berdasarkan subjek bisnis atau topik tertentu, seperti penjualan, keuangan, atau inventaris. Pendekatan ini memungkinkan pengguna untuk fokus pada data yang relevan dengan kebutuhan analisis mereka.
3. Orientasi Waktu: Datawarehouse menyimpan data historis untuk mendukung analisis tren dan pola dari waktu ke waktu. Hal ini memungkinkan pengguna untuk melacak perubahan dan evolusi dalam data bisnis mereka.
4. Model Data: Model data adalah struktur yang digunakan untuk mengatur data dalam datawarehouse. Dua model data yang umum digunakan dalam datawarehouse adalah model bintang dan model snowflake. Model bintang memiliki tabel fakta sentral yang terhubung dengan tabel dimensi di sekelilingnya. Model snowflake adalah variasi dari model bintang di mana dimensi dapat terbagi menjadi beberapa tabel yang terhubung.
5. Metadata: Metadata adalah informasi deskriptif tentang data yang disimpan dalam datawarehouse. Metadata mencakup informasi seperti definisi data, struktur data, dan sumber data. Metadata membantu pengguna dalam memahami dan mengakses data yang disimpan dalam datawarehouse.
6. Ekstraksi, Transformasi, dan Load (ETL): Proses ETL digunakan untuk mengumpulkan data dari sumber yang berbeda, mentransformasikannya ke dalam format yang sesuai untuk penyimpanan dalam datawarehouse, dan memuatnya ke dalam datawarehouse. ETL adalah tahap yang kritis dalam mempersiapkan data untuk analisis.
7. Akses dan Analisis: Datawarehouse menyediakan akses yang mudah bagi pengguna untuk menganalisis data menggunakan berbagai alat analisis seperti OLAP (Online Analytical Processing), data mining, dan laporan. Pengguna dapat menjalankan kueri kompleks dan melakukan analisis interaktif terhadap datawarehouse.
8. Skalabilitas dan Kinerja: Datawarehouse harus dirancang untuk mendukung pertumbuhan data yang besar dan menyediakan kinerja yang cepat dalam mengakses dan menganalisis data. Ini melibatkan pemilihan teknologi yang tepat dan perencanaan infrastruktur yang baik.

ARSITEKTUR DIAGRAM

Arsikur Diagram yang existing yang digunakan pada Tugure Data Warehouse



Gambar diatas adalah Arsitektur Diagram secara umum. Dalam diagram terdapat beberapa bagian yaitu.

Data Source

Data source atau sumber data yang ditarik dari berbagai sumber aplikasi. Sumber data tersebut memiliki perbedaan cara dalam hal penarikan datanya. Berikut adalah sumber-sumber data yang digunakan saat ini.

- PostgreSQL
- SQL Server
- MongoDB
- Excel

Data Warehouse

1. Nifi

Fungsi dari Nifi adalah sebagai server untuk data integration, ETL tools dan Orchestrator. Berikut adalah spesifikasi server SVRU-DWH-DIS.

Category	Value
Server Name	SVRU-DWH-DIS
Server IP External	
Server IP Internal	192.168.0.178
Operating System	Ubuntu-22.04
Memory/RAM	64 GB
Processor	16 vCPU



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Disk	512 GB SSD
Main Function	Data Integration Nifi

2. PostgreSQL

Fungsi dari PostgreSQL adalah sebagai tempat penyimpanan Datasource dan Datamart dari Tugure Data Warehouse. Berikut adalah spesifikasi server **SVRU-DWH-DB**.

Category	Value
Server Name	SVRU-DWH-DB
Server IP External	
Server IP Internal	192.168.0.177
Operating System	Ubuntu-22.04
Memory/RAM	64 GB
Processor	16 vCPU
Disk	1024 GB SSD
Main Function	Database PostgreSQL

BI/ Visualization

Tugure Dashboard visualization terpasang pada server ini. Dimana nanti pengguna dapat melakukan pembuatan dashboard report secara mandiri pada aplikasi Tugure Dashboard yang sudah disediakan. Tugure Dashboard merupakan Apache Superset yang di install di server ini. Berikut adalah spesifikasi dari server **SVRU-DWH-DVS**.

Category	Value
Server Name	SVRU-DWH-DVS
Server IP External	
Server IP Internal	192.168.0.176
Operating System	Ubuntu-22.04
Memory/RAM	64 GB
Processor	16 vCPU
Disk	256 GB SSD
Main Function	Data Visualization Superset



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Ni-Fi

Apache NiFi adalah sebuah software yang bersifat open source yang bertujuan untuk penyerapan data yang sederhana, kuat, dan andal yang dapat memproses dan mendistribusikan data antar berbagai sistem dan database. Pada project Tugure Data Warehouse, Apache NiFi digunakan untuk melakukan penarikan data staging dari database PostgreSQL, MongoDB, SQLServer dan Flatfile yang selanjutnya akan disimpan pada database PostgreSQL. Apache NiFi pada project Tugure Data Warehouse dapat diakses pada laman <https://192.168.0.178:8443/nifi/>. Job pada Apache NiFi berjalan otomatis sesuai dengan penjadwalan yang ditetapkan.

- Berikut merupakan tampilan awal dari laman <https://192.168.0.178:8443/nifi/>

The screenshot shows the Apache NiFi login interface. At the top, the browser's address bar displays "Not secure https://192.168.0.178:8443/nifi/login". Below the address bar, there are links for "LN", "Bookmarks", and "Work". The main page has a large blue water droplet icon on the left. In the center, there is a "Log In" button. Below it, there are two input fields: one for "User" containing "user" and another for "Password" containing "password". At the bottom right of the form area, there is a "LOG IN" button.



1.0

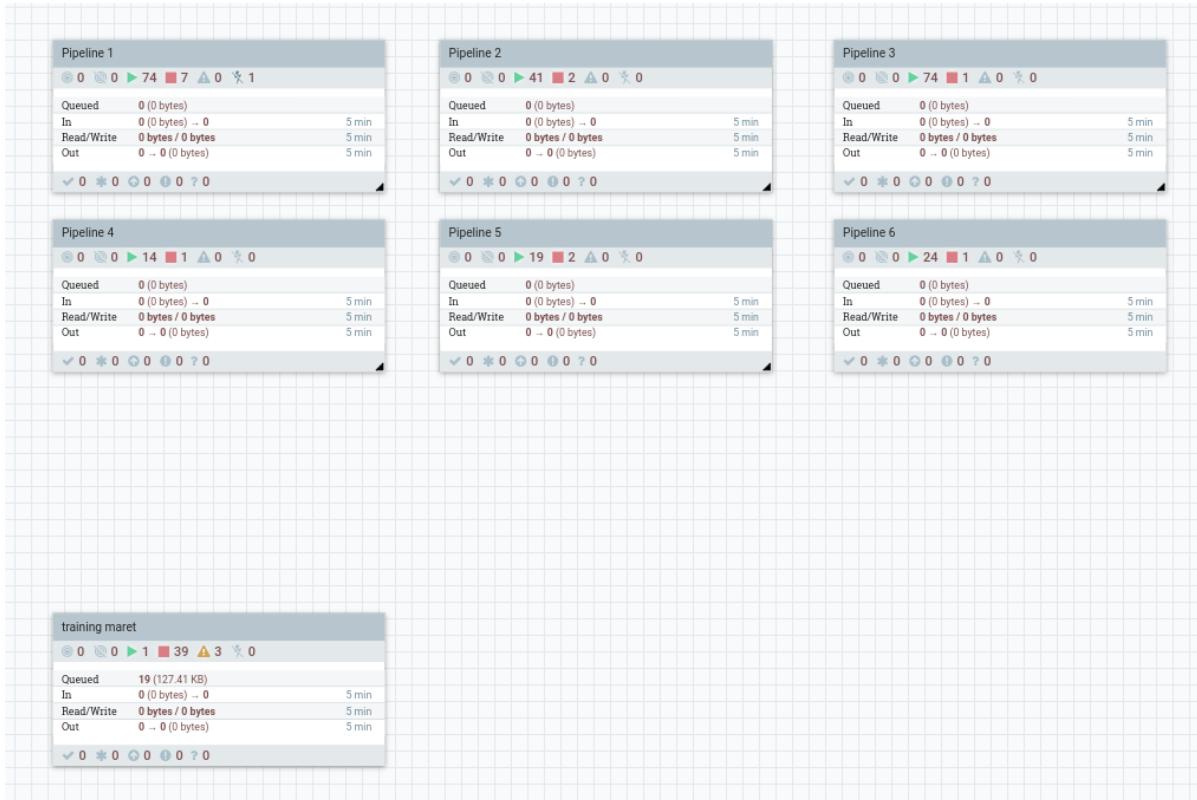
TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



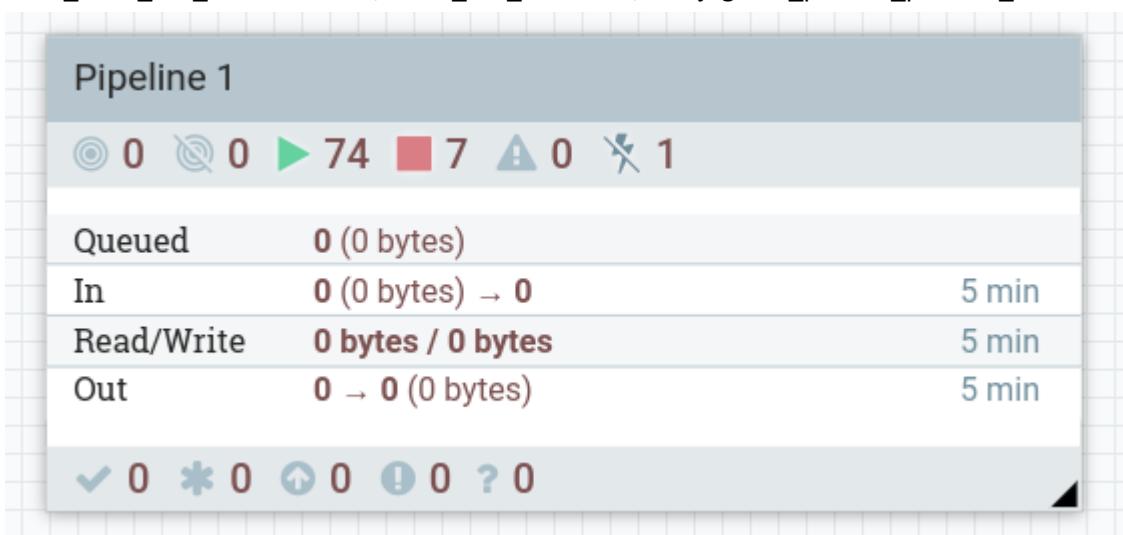
23 Februari 2024

- Berikut merupakan tampilan apabila sudah melakukan login ke Apache NiFi

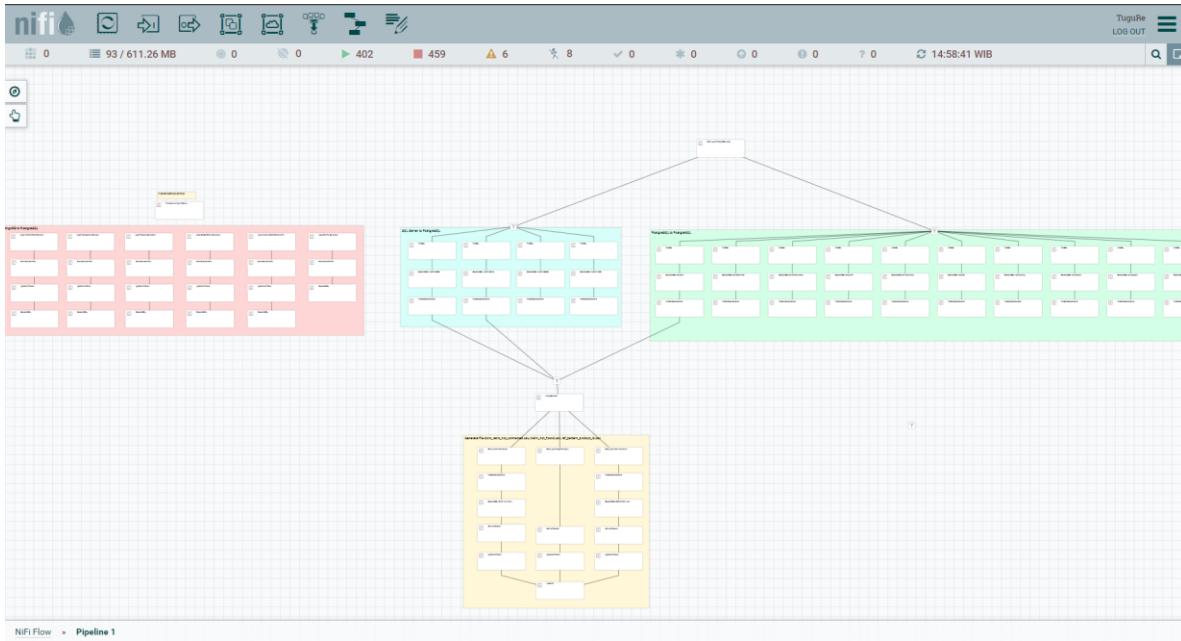


Pipeline 1

Pada Process Group Pipeline 1 berisi Processor untuk melakukan penarikan data dari beberapa source data Tugure seperti SQL Server, PostgreSQL, MongoDB yang kemudian disimpan di database PostgreSQL yang berjalan setiap bulan di tanggal 21 jam 01.00 WIB. Adapun dari Process Group Pipeline 1 ini akan menghasilkan beberapa file dengan extention .csv jika ada, yaitu file dcno_retro_not_connected.csv, claim_not_found.csv, dan juga ref_pattern_product_id.csv

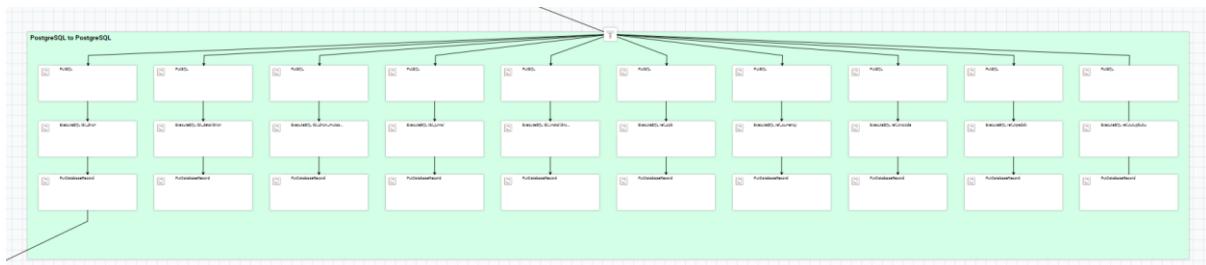


Process Group Pipeline 1 adalah sebagai berikut:



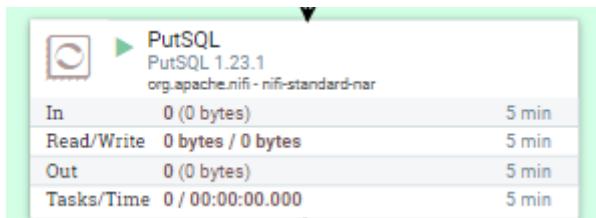
Di dalam Process Group Pipeline 1 berisi proses-proses sebagai berikut:

1. Proses ingestion source data PostgreSQL



Processor yang digunakan untuk penarikan data dari database PostgreSQL adalah sebagai berikut:

- PutSQL



Processor ini digunakan untuk mengeksekusi perintah SQL ke database PostgreSQL DWH, perintah SQL nya untuk menghapus (truncate) data dari tabel di database PostgreSQL DWH. Berikut konfigurasi di dalam processor PutSQL:



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Processor Details | PutSQL 1.23.1

Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
JDBC Connection Pool	DBCPConnectionPool - tugure_dwh
SQL Statement	truncate table staging.admnf2000
Support Fragmented Transactions	true
Database Session AutoCommit	false
Transaction Timeout	No value set
Batch Size	100
Obtain Generated Keys	false
Rollback On Failure	false

- ExecuteSQL

ExecuteSQL tbl_dncn
ExecuteSQL 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengeksekusi perintah SQL ke database PostgreSQL sumber dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

Berikut konfigurasi di dalam processor ExecuteSQL:

Processor Details | ExecuteSQL 1.23.1

Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Database Connection Pooling Service	DBCPConnectionPool - dbtugure
SQL Pre-Query	No value set
SQL select query	select * from dbo.admnf2000
SQL Post-Query	No value set
Max Wait Time	0 seconds
Normalize Table/Column Names	false
Use Avro Logical Types	false
Compression Format	NONE
Default Decimal Precision	10
Default Decimal Scale	0
Max Rows Per Flow File	0
Output Batch Size	0



1.0

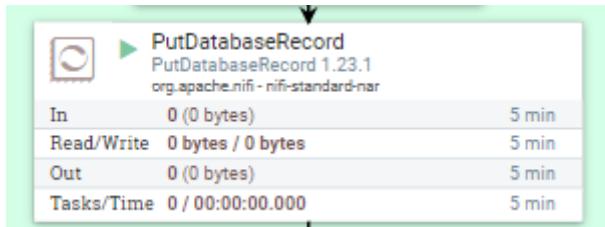
TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

- PutDatabaseRecord



Processor ini digunakan untuk menyimpan data (flowfile) ke dalam tabel di database PostgreSQL DWH.

Berikut konfigurasi di dalam processor PutDatabaseRecord:

Processor Details | PutDatabaseRecord 1.23.1

▶ Running STOP & CONFIGURE

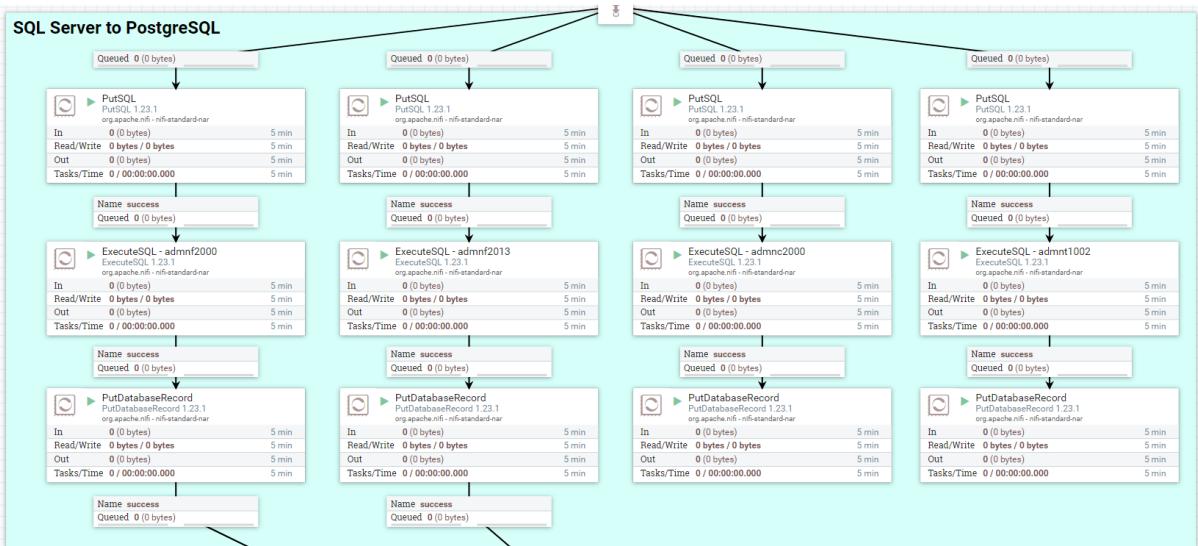
SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Record Reader	AvroReader
Database Type	PostgreSQL
Statement Type	INSERT
Data Record Path	No value set
Database Connection Pooling Service	DBCPConnectionPool - tugure_dwh
Catalog Name	No value set
Schema Name	staging
Table Name	admnf2000
Translate Field Names	true
Unmatched Field Behavior	Ignore Unmatched Fields
Unmatched Column Behavior	Ignore Unmatched Columns
Quote Column Identifiers	false

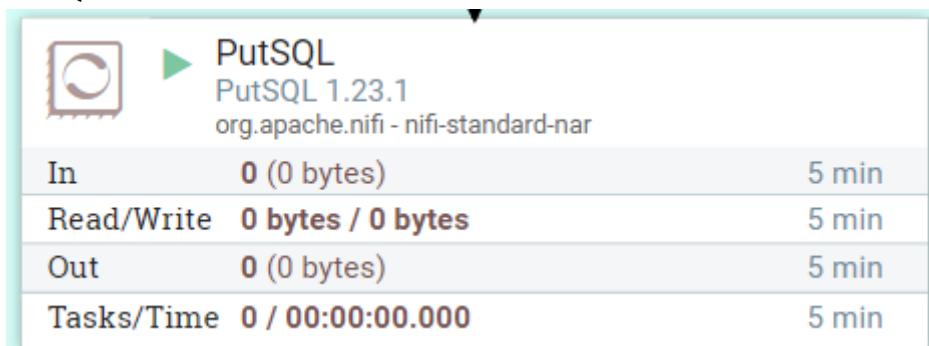
pk

2. Proses ingestion source data SQL Server



Processor yang digunakan untuk penarikan data dari source data SQL Server adalah sebagai berikut:

- PutSQL



Processor ini digunakan untuk mengeksekusi perintah SQL ke database PostgreSQL DWH, perintah SQL nya untuk menghapus (truncate) data dari tabel di database PostgreSQL DWH. Berikut konfigurasi di dalam processor PutSQL:



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Processor Details | PutSQL 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
JDBC Connection Pool	DBCPConnectionPool - tugure_dwh
SQL Statement	truncate table staging.admnf2000
Support Fragmented Transactions	true
Database Session AutoCommit	false
Transaction Timeout	No value set
Batch Size	100
Obtain Generated Keys	false
Rollback On Failure	false

- ExecuteSQL

ExecuteSQL - admnf2000
ExecuteSQL 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengeksekusi perintah SQL ke database SQL Server source data dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

Berikut konfigurasi di dalam processor ExecuteSQL:



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

Processor Details | ExecuteSQL 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Database Connection Pooling Service	DBCPConnectionPool - dbtugure
SQL Pre-Query	No value set
SQL select query	select * from dbo.admnf2000
SQL Post-Query	No value set
Max Wait Time	0 seconds
Normalize Table/Column Names	false
Use Avro Logical Types	false
Compression Format	NONE
Default Decimal Precision	10
Default Decimal Scale	0
Max Rows Per Flow File	0
Output Batch Size	0

OK

- PutDatabaseRecord

▶ PutDatabaseRecord
PutDatabaseRecord 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel di database DWH.

Berikut konfigurasi di dalam processor PutDatabaseRecord:

Processor Details | PutDatabaseRecord 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Record Reader	AvroReader
Database Type	PostgreSQL
Statement Type	INSERT
Data Record Path	No value set
Database Connection Pooling Service	DBCPConnectionPool - tugure_dwh
Catalog Name	No value set
Schema Name	staging
Table Name	admnf2000
Translate Field Names	true
Unmatched Field Behavior	Ignore Unmatched Fields
Unmatched Column Behavior	Ignore Unmatched Columns
Quote Column Identifiers	false

OK

3. Proses ingestion source data MongoDB



Processor yang digunakan untuk penarikan data dari source data MongoDB adalah sebagai berikut:

- GetMongo

Load note-numbers Collection
GetMongo 1.23.1
org.apache.nifi - nifi-mongodb-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 byte	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengeksekusi perintah SQL ke database MongoDB sumber dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

Berikut konfigurasi di dalam processor GetMongo:



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Configure Processor | GetMongo 1.23.1

Stopped

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Client Service	MongoDBControllerService
Mongo URI	No value set
Mongo Database Name	tugure-facultative
Mongo Collection Name	note-numbers
SSL Context Service	No value set
Client Auth	REQUIRED
JSON Type	Standard JSON
Pretty Print Results JSON	True
Character Set	UTF-8
Query	{}
Query Output Attribute	No value set
Projection	No value set

CANCEL APPLY

- EvaluateJsonPath

EvaluateJsonPath
EvaluateJsonPath 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengekstraksi nilai dari data JSON dalam flowfile dan menyimpannya sebagai atribut FlowFile.

Berikut konfigurasi di dalam processor EvaluateJsonPath:



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Processor Details | EvaluateJsonPath 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Destination	flowfile-attribute
Return Type	json
Path Not Found Behavior	ignore
Null Value Representation	the string 'null'
cob_code	\$cob.code
cob_id	\$cob.id
cob_name	\$cob.name
currency	\$currency
document_number	\$document_number
id	\$_id
index	\$index
number	\$number

OK

- UpdateAttribute

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk memperbarui atribut-atribut pada setiap FlowFile yang melewati proses tersebut dalam aliran data (flowfile), seperti update format Date dan replace string.

Berikut konfigurasi di dalam processor UpdateAttribute:



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Processor Details | UpdateAttribute 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value	
Delete Attributes Expression	?	No value set
Store State	?	Do not store state
Stateful Variables Initial Value	?	No value set
Cache Value Lookup Cache Size	?	100
sub_cob_name	?	\$(sub_cob_name:replace("", ""))
timestamp_created_at	?	\$(timestamp_created_at:replace("T","").replace("Z",".000")...)
timestamp_updated_at	?	\$(timestamp_updated_at:replace("T","").replace("Z",".000")...)

ADVANCED OK

- ExecuteSQL

ExecuteSQL
ExecuteSQL 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengeksekusi perintah SQL insert ke tabel di database DWH.
Berikut konfigurasi di dalam processor ExecuteSQL:



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Processor Details | ExecuteSQL 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Database Connection Pooling Service	DBCPConnectionPool - tugure_dwh
SQL Pre-Query	No value set
SQL select query	INSERT INTO staging.note_numbers (...)
SQL Post-Query	No value set
Max Wait Time	0 seconds
Normalize Table/Column Names	false
Use Avro Logical Types	false
Compression Format	NONE
Default Decimal Precision	10
Default Decimal Scale	0
Max Rows Per Flow File	0
Output Batch Size	0

OK

Berikut query yang digunakan untuk menyimpan data ke PostgreSQL DWH:

```
INSERT INTO staging.note_numbers (
    id,
    transaction_id,
    status,
    cob_id,
    cob_name,
    cob_code,
    sub_cob_id,
    sub_cob_name,
    sub_cob_parent_id,
    sub_cob_parent_name,
    sub_cob_parent_code,
    currency,
    "type",
    source_id,
    source_name,
    source_type,
    source_code,
    source_email,
    source_phone,
    "index",
    "number",
    timestamp_created_at,
    timestamp_created_by_role,
    timestamp_updated_at,
    timestamp_updated_by_role,
    document_number
)
SELECT
CASE
    WHEN '${id}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${id}'
END as id,
CASE
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        WHEN '${transaction_id}' IN ('null' , '' , '[]') THEN NULL
        ELSE '${transaction_id}'
    END as transaction_id,
CASE
    WHEN '${status}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${status}'
END as status,
CASE
    WHEN '${cob_id}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${cob_id}'
END as cob_id,
CASE
    WHEN '${cob_name}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${cob_name}'
END as cob_name,
CASE
    WHEN '${cob_code}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${cob_code}'
END as cob_code,
CASE
    WHEN '${sub_cob_id}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${sub_cob_id}'
END as sub_cob_id,
CASE
    WHEN '${sub_cob_name}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${sub_cob_name}'
END as sub_cob_name,
CASE
    WHEN '${sub_cob_parent_id}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${sub_cob_parent_id}'
END as sub_cob_parent_id,
CASE
    WHEN '${sub_cob_parent_name}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${sub_cob_parent_name}'
END as sub_cob_parent_name,
CASE
    WHEN '${sub_cob_parent_code}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${sub_cob_parent_code}'
END as sub_cob_parent_code,
CASE
    WHEN '${currency}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${currency}'
END as currency,
CASE
    WHEN '${type}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${type}'
END as type,
CASE
    WHEN '${source_id}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${source_id}'
END as source_id,
CASE
    WHEN '${source_name}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${source_name}'
END as source_name,
CASE
    WHEN '${source_type}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${source_type}'
END as source_type,
CASE
```



1.0

TECHNICAL DOCUMENT

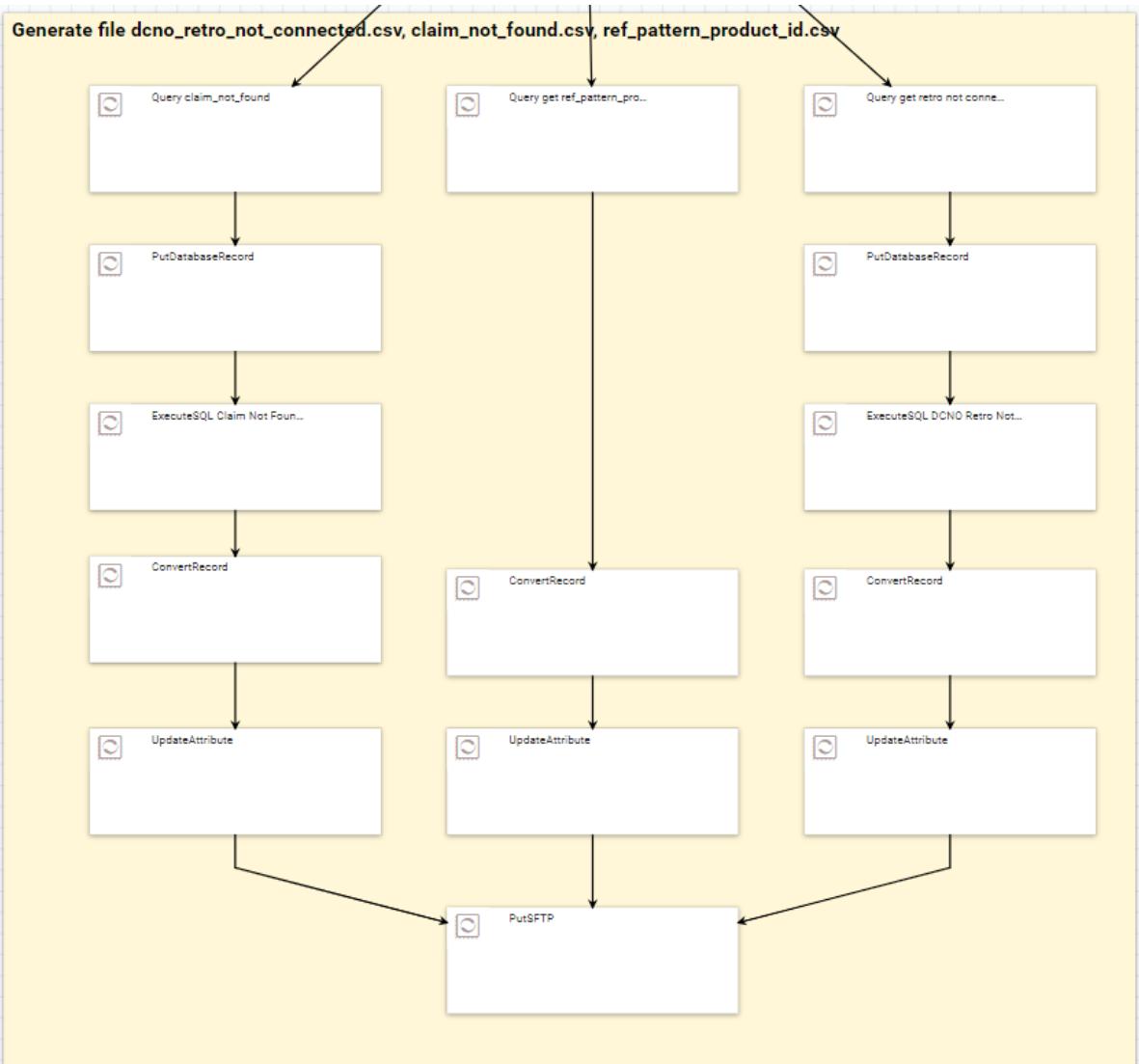
TUGURE DATA WAREHOUSE



23 Februari 2024

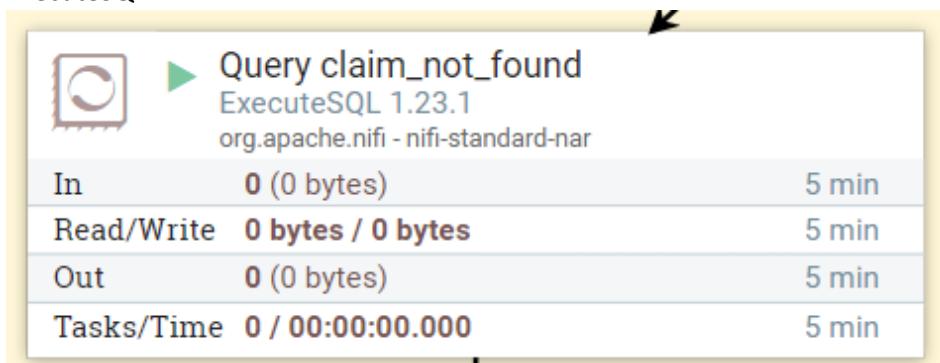
```
        WHEN '${source_code}' IN ('null' , '' , '[]') THEN NULL
        ELSE '${source_code}'
    END as source_code,
CASE
    WHEN '${source_email}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${source_email}'
END as source_email,
CASE
    WHEN '${source_phone}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${source_phone}'
END as source_phone,
CASE
    WHEN '${index}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${index}'
END::int as index,
CASE
    WHEN '${number}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${number}'
END as number,
CASE
    WHEN '${timestamp_created_at}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${timestamp_created_at}'
END::timestamp as timestamp_created_at,
CASE
    WHEN '${timestamp_created_by_role}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${timestamp_created_by_role}'
END::timestamp as timestamp_created_by_role,
CASE
    WHEN '${timestamp_updated_at}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${timestamp_updated_at}'
END::timestamp as timestamp_updated_at,
CASE
    WHEN '${timestamp_updated_by_role}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${timestamp_updated_by_role}'
END::timestamp as timestamp_updated_by_role,
CASE
    WHEN '${document_number}' IN ('null' , '' , '[]') THEN NULL
    ELSE '${document_number}'
END as document_number
```

4. Generate file CSV



Processor yang digunakan untuk pemrosesan data dan kemudian mengeluarkan file .csv sebagai outputnya adalah sebagai berikut:

- ExecuteSQL



Query claim_not_found
ExecuteSQL 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Processor ini digunakan untuk mengeksekusi perintah SQL untuk mendapatkan data claim yang tidak ditemukan kontrak sesinya untuk kemudian diambil hasilnya dan dijadikan aliran data (flowfile).

Berikut query yang digunakan untuk mendapatkan data claim yang tidak ditemukan kontrak sesinya:

```
with claims_mongodb as (
    select distinct
        register_claim,
        transaction->>'document_number' as document_number
    from staging.claims c
)
--select * from claims_mongodb;

,fadmnf2013 as (
    select nofileinduk, orc, nodebit
    from staging.admnf2013 where nodebit != ''
)
--select * from fadmnf2013;

,data_admnf2013 as (
    select nofileinduk, orc, min(nodebit) nodebit, count(*) cnt
    from fadmnf2013
    group by nofileinduk, orc
)
--select * from data_admnf2013;

,data_admnf2013_1 as (
    select nofileinduk, orc, nodebit from data_admnf2013 where cnt = 1
)
,
,data_admnf2013_2 as (
    select
        nofileinduk, orc,
        replace(nodebit,concat('.','.',split_part(nodebit,'.',4),'.','.'),'.00.') nodebit
    from data_admnf2013 where cnt > 1
)
,
,adm13 as (
    select * from data_admnf2013_1
    union all
    select * from data_admnf2013_2
)
,
,dncn_non_claim as (
    select
        dcno,
        dccurr,
        dcnoold,
        dcnofile,
        subcob,
        substring(dcno, 4,3) as kdtrx
    from staging.tbl_dncn where left(dcno,2) in ('01', '02')
)
--select * from dncn_non_claim;
,
,dncn_claim as (
    select
        dcnofile,
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
dcno,
subcob,
dccurr,
kdcob,
dcurref,
dcnoold,
dcclaim,
dcbebanclaim,
dcsalvage,
dcnet,
dcdateloss,
dctanggal,
substring(dcno,4,3) as kdtrx
from staging.tbl_dncn td
where
      left(dcno,2) in ('03','04')
)
--select * from dncn_claim;
,
claim_non_blips as (
    select
        dcno,
        kdcob,
        dccurr,
        dcurref,
        dcnoold,
        dctanggal,
        dcnofile,
        dcdateloss,
        sum(dcclaim) dcclaim,
        sum(dcbebanclaim) dcbebanclaim,
        sum(dcsalvage) dcsalvage,
        sum(dcnet) dcnet,
        substring(dcno, 4,3) as kdtrx,
        min(nofileinduk) nofileinduk
    from dncn_claim
    join staging.admnc2000 on dcurref = noclaim
    where
          left(dcno,2) in ('03','04')
          and nofileinduk != 0
    group by 1,2,3,4,5,6,7,8
)
--select count(*) from claim_non_blips;
,
link_claim_blips as (
    select
        a.dcno as dcno_claim,
        c.dcno as dcno_sesi,
        a.dcclaim,
        a.dcbebanclaim,
        a.dcsalvage,
        a.dctanggal,
        a.dcnoold,
        a.dcnofile,
        a.dcurref,
        a.dcnet,
        a.dcdateloss
    from dncn_claim a
    join claims_mongodb b on a.dcnofile = b.register_claim
```



1.0

TECHNICAL DOCUMENT

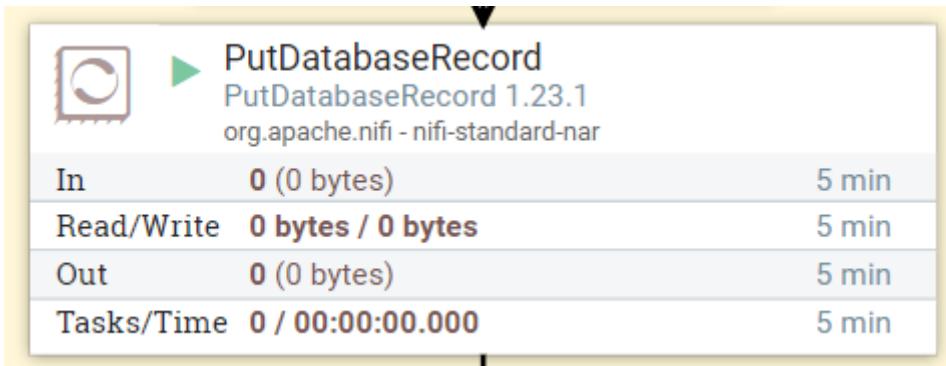
TUGURE DATA WAREHOUSE



23 Februari 2024

```
join dncn_non_claim c on b.document_number = c.dcnofile and a.subcob = c.subcob
and a.kdtrx = c.kdtrx and a.dccurr = c.dccurr
)
--select * from link_claim_bilips;
,
link_claim_non_bilips as (
    select
        a.dcno as dcno_claim,
        c.dcno as dcno_sesi,
        a.dcclaim,
        a.dcbebanclaim,
        a.dcsalvage,
        a.dctanggal,
        a.dcnoold,
        a.dcnofile,
        a.dcourref,
        a.dcnet,
        a.dcdateloss
    from
        claim_non_bilips a
    join adm13 b on a.nofileinduk = b.nofileinduk and a.dccurr = b.orc
    join dncn_non_claim c on b.nodebit = c.dcnoold and a.kdtrx = c.kdtrx
)
--select count(*) from link_claim_non_bilips;
,
all_link_claim as (
    select * from link_claim_bilips union all select * from link_claim_non_bilips
)
insert into staging.claim_not_found
select
    a.dcno,
    a.kdcob,
    case
        when a.dcno like '%-U%' then 'WebUpload'
        when a.dcno like '%-C%' and a.dcno not like '%-U%' then 'WebCredit'
        else 'BlipsOps'
    end source_of_data,
    a.dcclaim,
    a.dcbebanclaim,
    a.dcsalvage,
    a.dctanggal,
    a.dcnoold,
    a.dcnofile,
    a.dcourref,
    a.dcnet,
    a.dcdateloss,
    null as dcno_sesi
from
    staging.tbl_dncn a
left join all_link_claim b on
    a.dcno = b.dcno_claim
where
    left(a.dcno,2) in ('03', '04')
    and b.dcno_sesi is null
    and a.dcno not in (select dcno_claim from staging.claim_not_found);
```

- PutDatabaseRecord

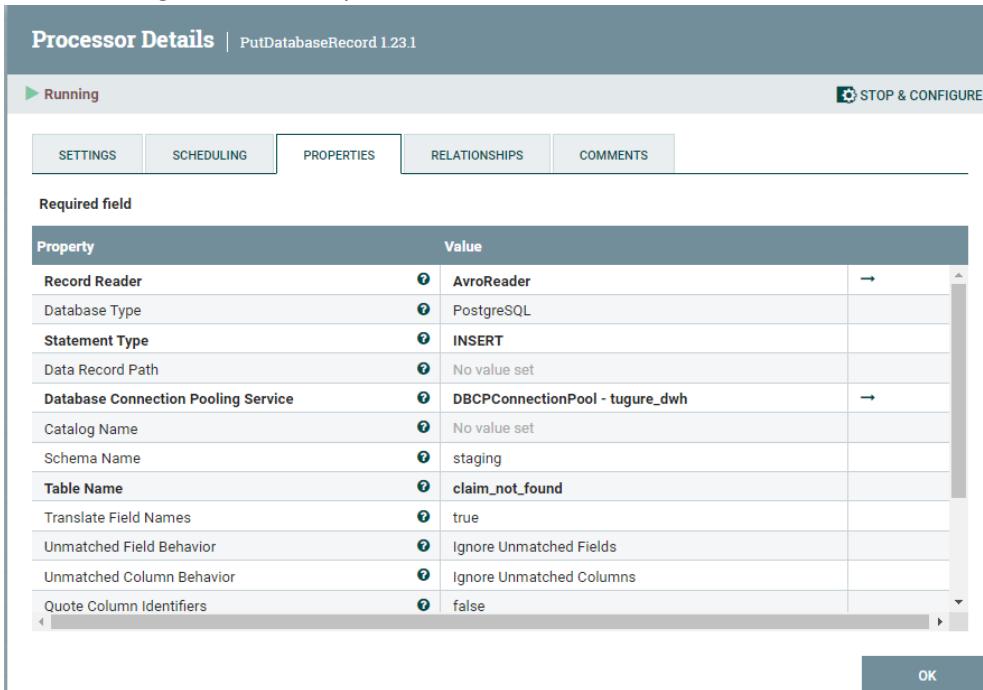


PutDatabaseRecord
PutDatabaseRecord 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel `claim_not_found` di database PostgreSQL DWH.

Berikut konfigurasi di dalam processor PutDatabaseRecord:



Processor Details | PutDatabaseRecord 1.23.1

▶ Running STOP & CONFIGURE

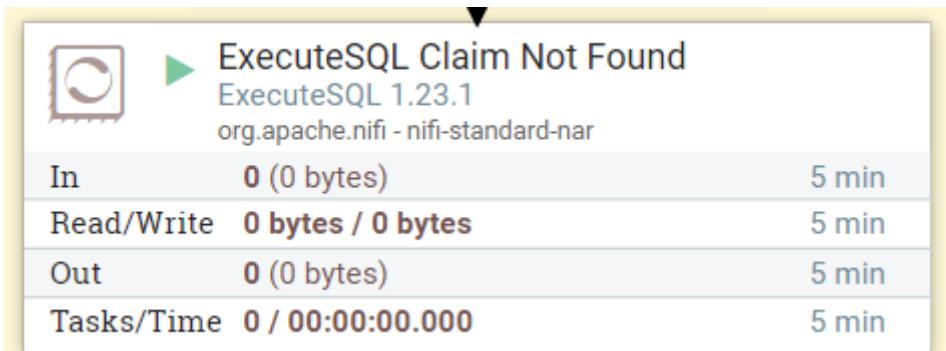
SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Record Reader	AvroReader
Database Type	PostgreSQL
Statement Type	INSERT
Data Record Path	No value set
Database Connection Pooling Service	DBCPConnectionPool - tugure_dwh
Catalog Name	No value set
Schema Name	staging
Table Name	claim_not_found
Translate Field Names	true
Unmatched Field Behavior	Ignore Unmatched Fields
Unmatched Column Behavior	Ignore Unmatched Columns
Quote Column Identifiers	false

OK

- ExecuteSQL



ExecuteSQL Claim Not Found
ExecuteSQL 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengeksekusi perintah SQL untuk mendapatkan data dari tabel `claim_not_found` dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Berikut konfigurasi di dalam processor ExecuteSQL:

Processor Details | ExecuteSQL 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Database Connection Pooling Service	DBCPConnectionPool - tugure_dwh
SQL Pre-Query	No value set
SQL select query	select ...
SQL Post-Query	No value set
Max Wait Time	0 seconds
Normalize Table/Column Names	false
Use Avro Logical Types	false
Compression Format	NONE
Default Decimal Precision	10
Default Decimal Scale	0
Max Rows Per Flow File	0
Output Batch Size	0

OK

- ConvertRecord

ConvertRecord
ConvertRecord 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk melakukan transformasi data dengan cara mengkonversi format data di aliran data (flowfile) dari format Afro menjadi CSV.

Berikut konfigurasi di dalam processor ConvertRecord:

Processor Details | ConvertRecord 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Record Reader	AvroReader
Record Writer	CSVRecordSetWriter
Include Zero Record FlowFiles	true



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

- UpdateAttribute

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk memperbarui atribut-atribut pada setiap FlowFile yang melewati proses tersebut dalam aliran data (flowfile), disini fungsinya untuk membuat atribut nama file yang akan di generate.

Berikut konfigurasi di dalam processor UpdateAttribute:

Property	Value
Delete Attributes Expression	No value set
Store State	Do not store state
Stateful Variables Initial Value	No value set
Cache Value Lookup Cache Size	100
filename	claim_not_found.csv

- PutSFTP

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengunggah (upload) file dari aliran data (flowfile) ke sebuah server SFTP (SSH File Transfer Protocol).

Berikut konfigurasi di dalam processor PutSFTP:



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

Processor Details | PutSFTP 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

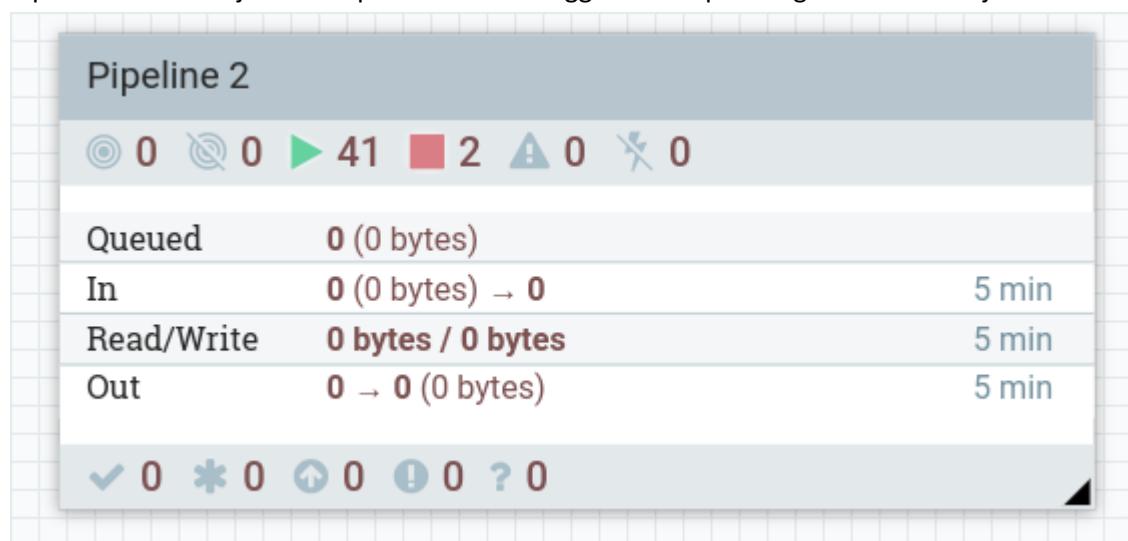
Required field

Property	Value
Hostname	192.168.0.178
Port	22
Username	tuguredwh
Password	Sensitive value set
Private Key Path	No value set
Private Key Passphrase	No value set
Remote Path	/nifi/data_source/generated_files/
Create Directory	false
Disable Directory Listing	false
Batch Size	500
Connection Timeout	30 sec
Data Timeout	30 sec

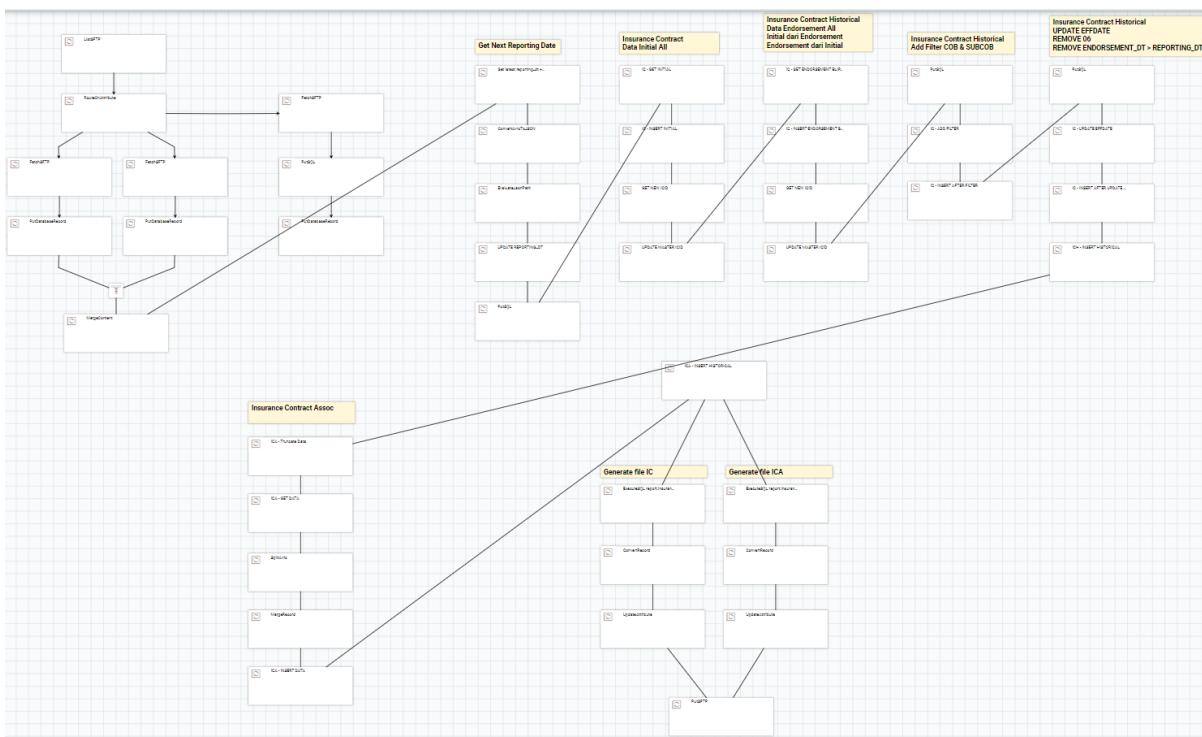
OK

Pipeline 2

Pada Process Group Pipeline 2 berisi Processor untuk melakukan ingestion terhadap beberapa file yang dihasilkan pada Process Group Pipeline 1 dan telah dilengkapi datanya oleh Tugure serta melakukan pembuatan data insurance contract dan insurance contract assoc dimana data tersebut akan disimpan di database PostgreSQL dan dikeluarkan dalam bentuk file dengan extension .csv. Process Group Pipeline 2 akan berjalan setiap bulan mulai tanggal 22 sampai dengan akhir bulan jam 01.00 WIB.



Process Group Pipeline 1 adalah sebagai berikut:



Di dalam Process Group Pipeline 1 berisi proses-proses sebagai berikut:



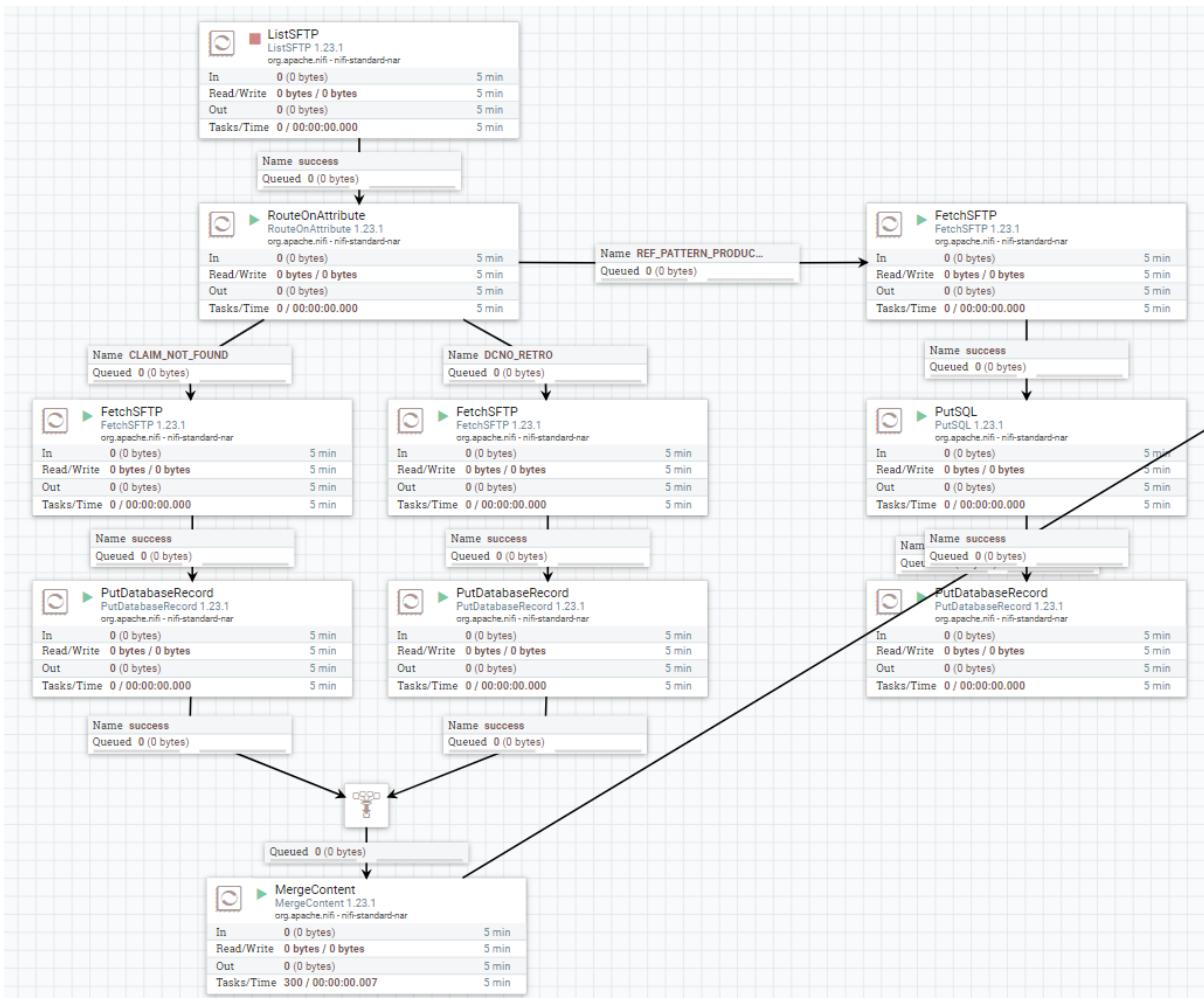
TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0

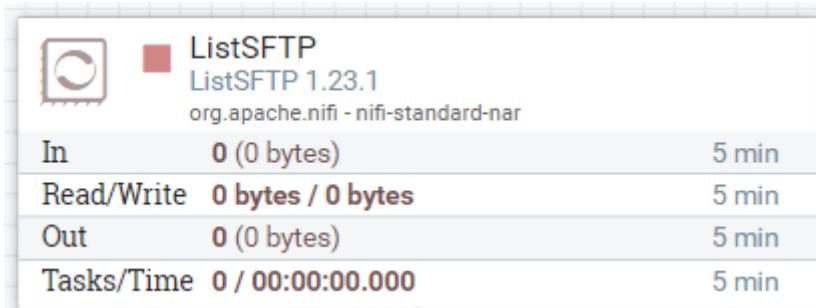
23 Februari 2024

1. Proses ingestion file `dcno_retro_not_connected.csv`, `claim_not_found.csv`, `danref pattern product id.csv`



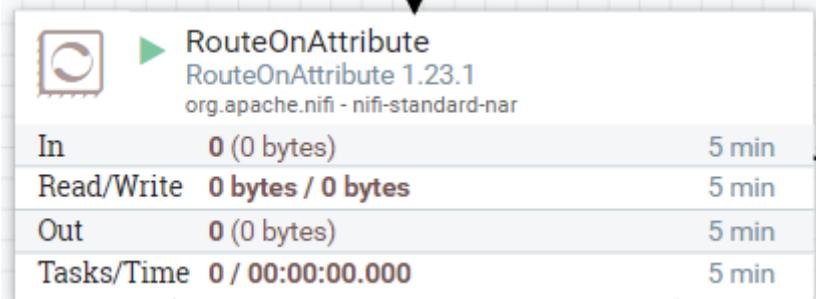
Processor yang digunakan untuk ingestion file dcno_retro_not_connected.csv, claim_not_found.csv, dan ref_pattern_product_id.csv adalah sebagai berikut:

- #### - ListSFTP



Processor ini digunakan untuk mendapatkan daftar file dari sebuah server SFTP (SSH File Transfer Protocol), untuk mendapatkan file dcno_retro_not_connected.csv, claim_not_found.csv, dan ref_pattern_product_id.csv

- RouteOnAttribute

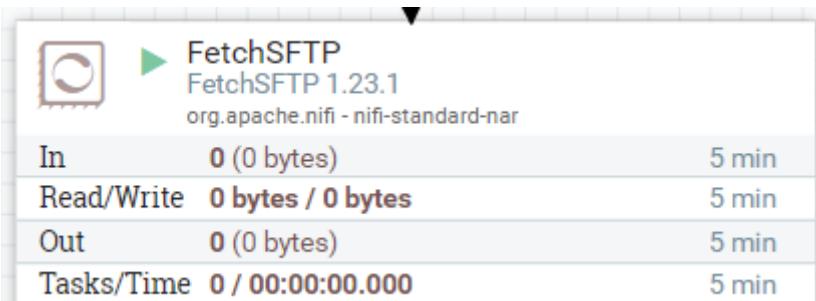


Processor ini digunakan untuk membagi aliran data (flowfile) menjadi cabang-cabang yang berbeda berdasarkan nilai atribut dari setiap FlowFile.



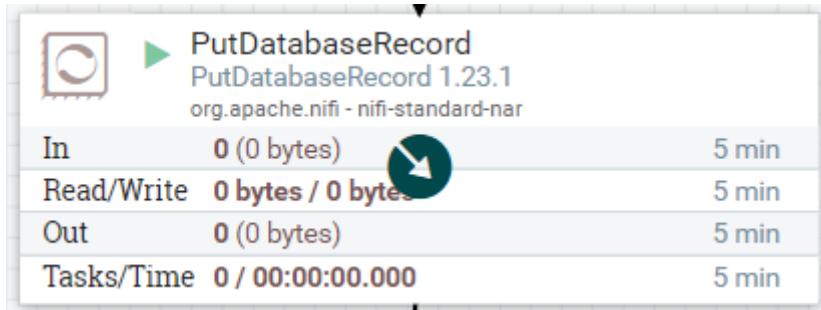
disini processor membagi flowfile menjadi 3 yaitu CLAIM_NOT_FOUND, DCNO_RETRO dan REF_PATTERN_PRODUCT_ID.

- FetchSFTP



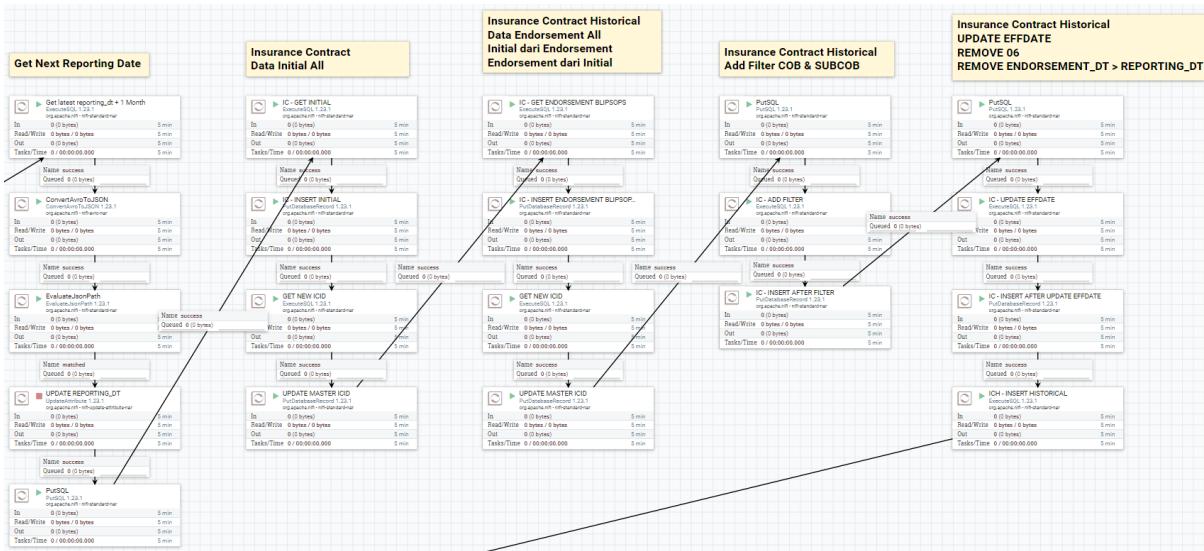
Processor ini digunakan untuk mengunduh (fetch) file dari server SFTP (SSH File Transfer Protocol), mengambil datanya kemudian datanya dijadikan aliran data (flowfile).

- PutDatabaseRecord



Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel di database PostgreSQL DWH.

2. Proses Insurance Contract



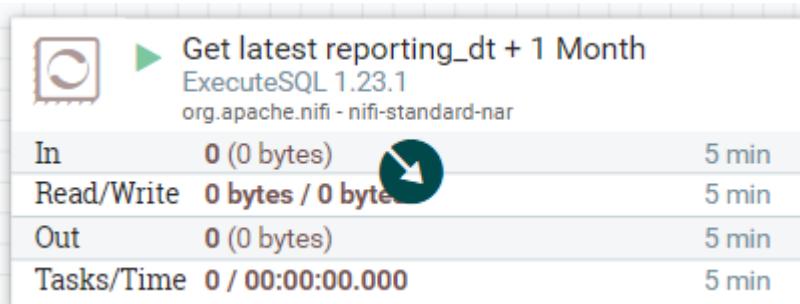
Step dalam proses insurance_contract adalah sebagai berikut:

a. Get Next Reporting Date

Pada proses ini akan menghasilkan nilai dari reporting_dt selanjutnya, berdasarkan data historical insurance_contract.

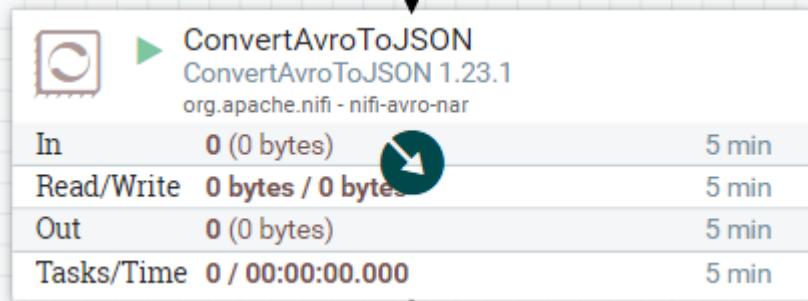
Processor yang digunakan untuk step adalah sebagai berikut:

- ExecuteSQL



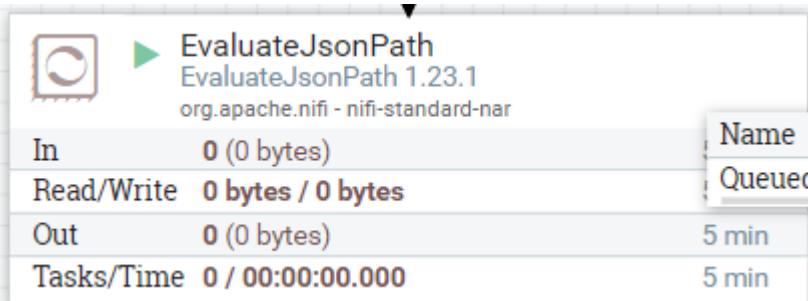
Processor ini digunakan untuk mengeksekusi perintah SQL untuk mendapatkan nilai reporting_dt selanjutnya dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

- ConvertAvroToJson



Processor ini digunakan untuk mengonversi data yang disimpan dalam format Avro menjadi format JSON.

- EvaluateJsonPath

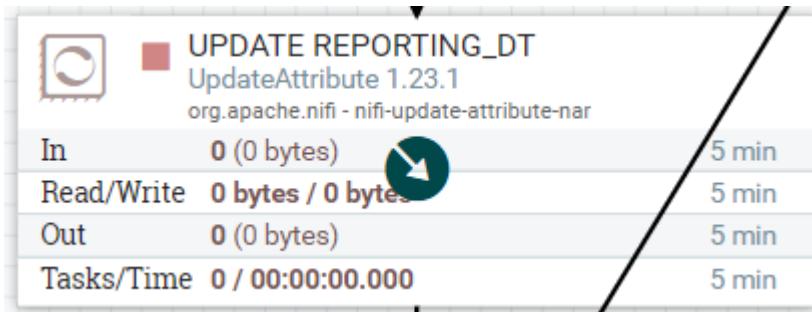


Processor ini digunakan untuk mengekstraksi nilai dari data JSON dan menyimpannya sebagai atribut FlowFile.

SETTINGS	SCHEDULING	PROPERTIES	RELATIONSHIPS	COMMENTS														
Required field																		
<table border="1"><thead><tr><th>Property</th><th>Value</th></tr></thead><tbody><tr><td>Destination</td><td>flowfile-attribute</td></tr><tr><td>Return Type</td><td>auto-detect</td></tr><tr><td>Path Not Found Behavior</td><td>ignore</td></tr><tr><td>Null Value Representation</td><td>empty string</td></tr><tr><td>reporting_dt</td><td>\$next_reporting_dt</td></tr><tr><td>year_reporting_dt</td><td>\$year_next_reporting_dt</td></tr></tbody></table>					Property	Value	Destination	flowfile-attribute	Return Type	auto-detect	Path Not Found Behavior	ignore	Null Value Representation	empty string	reporting_dt	\$next_reporting_dt	year_reporting_dt	\$year_next_reporting_dt
Property	Value																	
Destination	flowfile-attribute																	
Return Type	auto-detect																	
Path Not Found Behavior	ignore																	
Null Value Representation	empty string																	
reporting_dt	\$next_reporting_dt																	
year_reporting_dt	\$year_next_reporting_dt																	
SETTINGS	SCHEDULING	PROPERTIES	RELATIONSHIPS	COMMENTS														

Dalam processor ini attribut yang di set adalah reporting_dt dan year_reporting_dt yang berasal dari data flowfile.

- UpdateAttribute



Processor ini digunakan untuk memperbarui atau menambahkan atribut-atribut pada setiap FlowFile yang melewati prosesor tersebut dalam aliran data (flowfile).



Required field

Property	Value
Delete Attributes Expression	No value set
Store State	Do not store state
Stateful Variables Initial Value	No value set
Cache Value Lookup Cache Size	100
insurance_contract	insurance_contract
insurance_contract_assoc	insurance_contract_assoc
insurance_contract_assoc_historical	insurance_contract_assoc_historical
insurance_contract_historical	insurance_contract_historical
insurance_contract_step1	insurance_contract_step1
insurance_contract_step2	insurance_contract_step2
reporting_dt	2023-03-31
year_reporting_dt	2023

Dalam processor ini attribut yang di set adalah sebagai berikut:

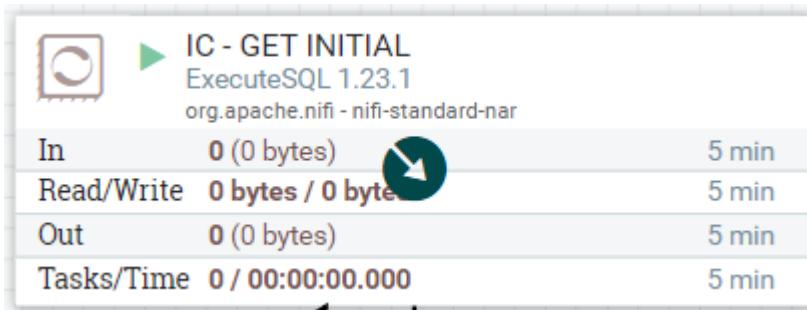
- insurance_contract (nama target tabel insurance_contract)
- insurance_contract_historical (nama target tabel insurance_contract_historical)
- insurance_contract_assoc (nama target tabel insurance_contract_assoc)
- insurance_contract_assoc_historical (nama target tabel insurance_contract_assoc_historical)
- insurance_contract_step1 (tabel temporary step 1)
- insurance_contract_step2 (tabel temporary step 2)
- reporting_dt (update nilai reporting_dt saat running manual)
- year_reporting_dt (update nilai year_reporting_dt saat running manual).

b. Get Initial Data

Pada proses ini akan menghasilkan data initial dari tbl_dncn untuk kemudian disimpan kedalam tabel insurance_contract.

Processor yang digunakan untuk step adalah sebagai berikut:

- ExecuteSQL



Processor ini digunakan untuk mengeksekusi perintah SQL untuk mendapatkan data initial dari `tbl_dncn` dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

Berikut query yang digunakan untuk menghasilkan data initial:

```
--1. INITIAL DATA
with set_cancelation as (
  select
    a.dcno, a.dcnoold, a.cedkd, a.kdcob, a.dcmulticob, a.dctipedok,
    a.dcbisnisunit, a.dctrxcode, a.dckob, a.dctanggal,
    a.dcnofile, a.dcpolicyno, a.dlakindofpayments, a.dcjenis,
    a.dcasuransi, a.dccurr, a.dcgrosspremium, a.dccommission,
    a.dcinfee, a.dcbrokerage, a.dccclaim, a.dcbebanclaim,
    a.dcpocommission, a.dcovrcommission, a.dcfleetdiskon, a.dcshare,
    a.dcpremainot, a.dctax, a.dcsalvage, a.dcnet, a.dcduedate, a.dcwpc,
    a.dcpriodestart, a.dcpriodeend, a.dcinuk, a.dciinsured,
    UPPER(TRIM(a.dciinsuredname)) as dciinsuredname, a.dcnoda,
    a.dcdateloss, a.dcuyear, a.dcsao, a.dcketerangan,
    a.dcconfreff, a.dcremarkbind, a.dcrislip, a.dcourref,
    a.dcreinstate, a.dcauth, a.dcjnkode, a.dcnetygditerima,
    a.dcnetygdibayar, a.dcnnetttotalpem, a.dcneterimig, a.dcnbalance,
    a.dctsmapping, a.dcnotemapping, a.dcsqllmapping,
    a.dctsjsurnal, a.dcpriodesoa, a.dcpriodsoquarter, a.dcpriodsoath,
    a.dctsvoxpro, a.dcsuninsured, a.dcgppremium,
    a.dcpcentpremi, a.dcpcentcomm, a.create_user, a.create_date,
    a.update_user, a.update_date, a.create_user_akseptasi,
    a.dcfasttrack, a."source", a.dcuunit, a.dcretro, a.subcob, a.dchp,
    a.qq_note, a.policy_holder, a.policy_holder_code,
    a.dcpriodestartins, a.dcpriodeendins, a.dcnnotaretro,
    'Cancellation' as dcstatus,
    a.dcvat, a.dcnoinvoice, a.dcnokontrak, a.dendorsement_type,
    a.dcnnameofrisk, a.dcpolicy_type, a.dwh_created_date
  from staging.tbl_dncn a
  left join staging.tbl_dncn b on trim(SPLIT_PART(a.dcinuk, '|', 1)) = b.dcno
  where 1=1
    and a.dcinuk is not null and a.dcinuk != ''
    and (a.dcstatus is null or a.dcstatus = '')
    and a.dcnofile = b.dcnofile
    and (
      a.dcgrosspremium
      *
      case
        when left(a.dcno,2) in ('07','08','05','06')
        then 1
        when left(a.dcno,2) in ('01','03') then 1
        when left(a.dcno,2) in ('02','04') then -1
        else 0
      end
    )
)
then 1
else 0
end
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
+  
b.dcgrosspremium  
*  
case  
when left(b.dcno,2) in ('07','08','05','06')  
then 1  
when left(b.dcno,2) in ('01','03') then 1  
when left(b.dcno,2) in ('02','04') then -1  
else 0  
end  
) = 0  
)  
--select dcinduk,dcstatus,* from set_cancelation where dcnofile =  
'FAC202101180028';  
,  
tbl_dncn_all as (  
select  
a.dcno, a.dcnold, a.cedkd, a.kdcob, a.dcmulticob, a.dctipedok,  
a.dcbisnisunit, a.dctrxcode, a.dckob,  
a.dctanggal, a.dcnofile, a.dcpolicyno, a.dlakindofpayments, a.dcjenis,  
a.dcasuransi, a.dccurr,  
a.dcgrosspremium, a.dccommission, a.dciintfee, a.dcbrokerage,  
a.dccclaim, a.dcbebanclaim, a.dcpocommission,  
a.dcovrcommision, a.dcfleetdiskon, a.dcsshare, a.dcpremainot, a.dctax,  
a.dcsalvage, a.dcnet, a.dcduedate,  
a.dcwpc, a.dcpriodestart, a.dcpriodeend, a.dcinduk, a.dciinsured,  
UPPER(TRIM(a.dciinsuredname)) as dciinsuredname,  
a.dcnoda, a.dcdateloss, a.dcuwyear, a.dcsoa, a.dcketerangan,  
a.dcconfreff, a.dcremarkbind, a.dcrislip, a.dcourref,  
a.dcreinstatement, a.dcauth, a.dcjnkode, a.dcnetygditerima,  
a.dcnetygdibayar, a.dcnnetttotalpem, a.dcneterimig,  
a.dcnetsbalance, a.dctsmapping, a.dcnatemapping, a.dcsq1mapping,  
a.dctsjsurnal, a.dcpriodesoa,  
a.dcpriodsoquarter, a.dcpriodsoath, a.dctsvoxpro, a.dcsuninsured,  
a.dcgppremium, a.dcpcentpremi,  
a.dcpcentcomm, a.create_user, a.create_date, a.update_user,  
a.update_date, a.create_user_akseptasi,  
a.dcfasttrack, a."source", a.dcuunit, a.dcretro, a.subcob, a.dchp,  
a.qq_note, a.policy_holder,  
a.policy_holder_code, a.dcpriodestartins, a.dcpriodeendins,  
a.dcnnotaretro, a.dcsstatus, a.dcvat,  
a.dcnoinvoice, a.dcnokontrak, a.dendorsement_type, a.dcnameofrisk,  
a.dcpolicy_type, a.dwh_created_date  
from staging.tbl_dncn a  
left join set_cancelation b on a.dcno = b.dcno  
where b.dcno is null  
union all  
select * from set_cancelation  
)  
--select dcinduk,dcstatus,dcgrosspremium,* from tbl_dncn_all where dcnofile =  
'FAC202101180028';  
,  
tbl_dncn_category as (  
select *,  
case when dcno like '%-U%' then 'WebUpload'  
when dcno like '%-C%' and dcno not like '%-U%' then 'WebCredit'  
else 'BlipsOps'  
end as category  
from tbl_dncn_all a  
where 1=1
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        and left(dcno,2) not in ('03','04','07','08') -- claim (03,04) & soa
treaties(07,08) di exclude
        and (
                (dcperiodestart      <=      '${year_reporting_dt}-01-01'      and
dcperiodeend >= '${year_reporting_dt}-01-01')
                or
                (dcperiodestart      >      '${year_reporting_dt}-01-01'      and
dcperiodestart <= '${reporting_dt}'
                        -- and dcperiodeend >= '${reporting_dt}'
                )
        )
),
initial_BlipsOps as (
    select * from tbl_dncn_category
    where category = 'BlipsOps' and (
        trim(dcstatus) in ('New', 'Replacement')
        or trim(dcstatus) = ''
        or trim(dcstatus) is null
        or trim(dcinduk) is null
        or trim(dcinduk) = ''
        or (trim(dcinduk) like '%|%' and trim(SPLIT_PART(dcinduk, '|', 1)) is
null)
        or (trim(dcinduk) like '%|%' and trim(SPLIT_PART(dcinduk, '|', 1)) =
'')
    )
),
initial_WebUpload as (
    select * from tbl_dncn_category
    where category = 'WebUpload' and dcnofile not like '%.E' and dcnofile not like
'%.CL'
),
initial_WebCredit as (
    select * from tbl_dncn_category
    where category = 'WebCredit'
),
all_initial as (
    select * from initial_BlipsOps
    union all
    select * from initial_WebUpload
    union all
    select * from initial_WebCredit
),
tbl_dncn_category_per_dcno as (
    select a.dcno, dcnofile, subcob,
    case
        when coalesce(b.kdcob, a.kdcob) = 'HE' then 'EN'
        when left(a.dcno,2) in ('05','06') then coalesce(b.kdcob,
a.kdcob, 'LF')
                else coalesce(b.kdcob, a.kdcob)
    end as kdcob,
    a.dccurr,
    case
        when b.dcno is not null then sum(coalesce(b.ddcpremium,0))
        else sum(coalesce(a.dcgrosspremium,0))
    end
    *
    case
        when left(a.dcno,2) in ('07','08','05','06') then 1
        when left(a.dcno,2) in ('01','03') then 1
        when left(a.dcno,2) in ('02','04') then -1
    end
)
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        else 0
    end      as prem_amt
from tbl_dncn_category a
left join staging.tbl_detaildncn b on a.dcno = b.dcno
group by a.dcno, b.dcno, a.kdcob, b.kdcob, a.dcnofile, a.subcob, a.dccurr
),
link_sesi_retro_dwh as (
    select
        a.dcno as dcno_sesi,
        a.dcnofile, a.kdcob, a.subcob, a.dccurr,
        string_agg(b.dcno,'|') dcno_retro,
        max(a.prem_amt) as prem_amt_sesi,
        sum(b.prem_amt) as prem_amt_retro
    from (
        select * from tbl_dncn_category_per_dcno a where a.dcno like '%K11%'
    ) a
    join (
        select * from tbl_dncn_category_per_dcno a where a.dcno like '%K51%'
    ) b on a.dcnofile = b.dcnofile and a.kdcob = b.kdcob and a.subcob = b.subcob
and a.dccurr = b.dccurr
        group by a.dcno, a.kdcob, b.kdcob, a.subcob, a.dccurr, a.dcnofile
),
link_sesi_retro_manual as (
    select a.*, b.prem_amt as prem_amt_sesi, c.prem_amt as prem_amt_retro from (
        select distinct dcno_sesi, dcno, dcnofile, kdcob, subcob, dccurr from
report.dcno_retro_not_connected
            where dcno_sesi is not null or dcno_sesi != ''
    ) a
    left join tbl_dncn_category_per_dcno b on a.dcno_sesi = b.dcno
    left join tbl_dncn_category_per_dcno c on a.dcno = c.dcno
),
link_sesi_retro as (
    select distinct
        coalesce(a.dcno_sesi,b.dcno_sesi) as dcno_sesi,
        coalesce(a.dcnofile,b.dcnofile) as dcnofile,
        coalesce(a.kdcob,b.kdcob) as kdcob,
        coalesce(a.subcob,b.subcob) as subcob,
        coalesce(a.dccurr,b.dccurr) as dccurr,
        coalesce(a.dcno_retro,b.dcno) as dcno_retro,
        coalesce(a.prem_amt_sesi,b.prem_amt_sesi) as prem_amt_sesi,
        coalesce(a.prem_amt_retro,b.prem_amt_retro) as prem_amt_retro
    from link_sesi_retro_dwh a
    full outer join link_sesi_retro_manual b on a.dcno_sesi = b.dcno_sesi and
a.dcno_retro = b.dcno
),
dcno_kob as (
    select distinct
        aa.dcno,
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
                else coalesce(gg.kdcob, aa.kdcob)
        end as kdcob,
        case
            when aa.dctrxcode in ('11','51') then '5'
            when aa.dctrxcode = '22' then
                case
                    when ii.jenis = 'SOA Life' then '4'
                    else

```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
case
then '1'
when nn.jenis = 'SOA Treaty QS'
when nn.jenis = 'SOA Treaty SPL'
when nn.jenis = 'SOA Treaty FACOB'
else '4'
end
end kob_cd,
case
when aa.dctrxcode in ('11','51') then 'FC'
when aa.dctrxcode = '22' then
case
when ii.jenis = 'SOA Life' then 'TN'
else
case
when nn.jenis = 'SOA Treaty QS'
when nn.jenis = 'SOA Treaty SPL'
when nn.jenis = 'SOA Treaty FACOB'
else 'TN'
end
else 'TN'
end as kob_id
from all_initial aa
left join staging.tbl_detaildncn gg on aa.dcno = gg.dcno
left join staging.ref_tipedok ii on aa.dctipedok = ii.code
left join staging.ref_tipedok nn on gg.ddckot = nn.code
),
latest_icid as (
select
coalesce(max(left(replace(insurance_contract_id,'-','0'), 5)::int),0)
latest_icid
from report.master_icid a
where date_part('YEAR',reporting_dt) = date_part('YEAR',
${reporting_dt}::date)
),
generate_icid_initial as (
select
'${reporting_dt}':date as reporting_dt,
'TRE' as entity_id,
coalesce(
uu.insurance_contract_id,
concat(
to_char((row_number() over(partition by
date_part('YEAR', ${reporting_dt}::date) order by aa.dcpriodestart asc)),
'fm00000'), -- Running Number
to_char(date_part('MONTH',
${reporting_dt}::date),'fm00'), -- Month
ff.kode_lokasi_afiliasi, -- Lokasi & Afiliasi
case
when coalesce(gg.kdcob, aa.kdcob) = 'HE' then
'EN'
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
when left(aa.dcno,2) in ('05','06') then
coalesce(gg.kdcob, aa.kdcob, 'LF')
else coalesce(gg.kdcob, aa.kdcob)
end, -- COB
coalesce(kk.sub_cob_id,oo.sub_cob_id,'00'), -- SUB COB

to_char(date_part('YEAR',coalesce(aa.dcperiodestart,'1960-01-01)::date)),
'fm0000'), -- Cohort
ff.jenis, -- Source of Business
coalesce(ee1.idinsured,ee2.idinsured), -- Insurred
to_char(ff.kode_ifrs::int,'fm000'), -- Ceding
case
when bb.sesiretro = 'SESI' then 'S'
when bb.sesiretro = 'RETRO' then 'R'
else '-'
end, -- Insurance type
rr.kob_cd, -- KOB
dd.coa_prefix -- Currency
)
) as insurance_contract_id,
aa.dcendorsement_type as endorsement_type_cd,
coalesce(aa.dcperiodestart,'1960-01-01)::date) as issue_dt,
case
when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
else coalesce(gg.kdcob, aa.kdcob)
end as product_line_id,
rr.kob_id,
concat(
coalesce(kk.sub_cob_id,oo.sub_cob_id,'00'),
'_',
rr.kob_id,
'_',
-- ((EXTRACT(MONTH FROM dcperiodeend) - EXTRACT(MONTH FROM
dcperiodestart)) + (12 * (EXTRACT(YEAR FROM aa.dcperiodeend) -
EXTRACT(YEAR FROM aa.dcperiodestart))))+1), 'mth',
case
when
(
(12 - EXTRACT(MONTH FROM dcperiodestart))
+
(EXTRACT(MONTH FROM dcperiodeend)) +
((EXTRACT(YEAR FROM dcperiodeend) -
EXTRACT(YEAR FROM dcperiodestart)) * 12) +
(case when EXTRACT(DAY FROM dcperiodeend) -
EXTRACT(DAY FROM dcperiodestart) > 0 then 1 else 0 end)
) = 0 then 1
else
(
(12 - EXTRACT(MONTH FROM dcperiodestart))
+
(EXTRACT(MONTH FROM dcperiodeend)) +
((EXTRACT(YEAR FROM dcperiodeend) -
EXTRACT(YEAR FROM dcperiodestart)) * 12) +
(case when EXTRACT(DAY FROM dcperiodeend) -
EXTRACT(DAY FROM dcperiodestart) > 0 then 1 else 0 end)
)
end, 'mth',
case
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
when rr.kob_id in ('TQ','TS') then
concat('_',upper(qq.periode_pelaporan))
when rr.kob_id in ('FC','TN') then
concat('_INSTALL',ss.n_installment,'_',tt.pattern)
else ''
end
) as product_id,
coalesce(ff.jenis,'') as source_of_business,
to_char(ff.kode_ifrs::int,'fm000') as ceding_id,
to_char(date_part('YEAR',coalesce(aa.dcp period start,'1960-01-01)::date)), 'fm0000') as cohort_id,
case
when bb.sesiretro = 'SESI' then 'N'
when bb.sesiretro = 'RETRO' then 'Y'
else '-'
end as ceded_flg,
case
when bb.sesiretro = 'SESI' then 'N'
else
case
when left(aa.dcno,2) in ('07','08') then
case
when gg.ddckot in ('15','16','17') then 'Y'
else 'N'
end
else 'Y'
end
end as reins_prop_cover_flg,
case
when bb.sesiretro = 'SESI' then 'N'
else
case
when bb."name" like '%Treaty%' then 'Y'
else 'N'
end
end as reins_treaty_flg,
null::date as endorsement_dt,
coalesce(aa.dcp period start,'1960-01-01)::date) as begin_cov_dt,
coalesce(aa.dcp period end,'1960-01-01)::date) as end_cov_dt,
aa.dccurr as currency_cd,
(
case
when gg.dcno is not null then
sum(coalesce(gg.ddc premium,0))
else sum(coalesce(aa.dcgross premium,0))
end
*
case
when left(aa.dcno,2) in ('07','08','05','06') then 1
when left(aa.dcno,2) in ('01','03') then 1
when left(aa.dcno,2) in ('02','04') then -1
else 0
end
) as prem_amt,
case
when bb.sesiretro = 'RETRO' then 0
when ll.dcno_sesi is not null then
(
case
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



1.0

23 Februari 2024

```
when gg.dcno is not null then
sum(coalesce(gg.ddcpremium,0))
else sum(coalesce(aa.dcgrosspremium,0))
end
*
case
when left(aa.dcno,2) in
('07','08','05','06') then 1
when left(aa.dcno,2) in ('01','03') then
when left(aa.dcno,2) in ('02','04') then
else 0
end
)
+
sum(l1.prem_amt_retro)
else
case
when gg.dcno is not null then
sum(coalesce(gg.ddcpremium,0))
else sum(coalesce(aa.dcgrosspremium,0))
end
*
case
when left(aa.dcno,2) in ('07','08','05','06')
then 1
when left(aa.dcno,2) in ('01','03') then 1
when left(aa.dcno,2) in ('02','04') then -1
else 0
end
end as base_alloc,
(
case
when gg.dcno is not null then
sum(coalesce(gg.ddcbrokerage,0))
else sum(coalesce(aa.dcbrokerage,0))
end
*
case
when left(aa.dcno,2) in ('07','08','05','06') then 1
when left(aa.dcno,2) in ('01','03') then -1
when left(aa.dcno,2) in ('02','04') then 1
else 0
end
) as comm_broker_amt,
(
case
when gg.dcno is not null then
sum(coalesce(gg.ddccommission,0))
else sum(coalesce(aa.dccommission,0))
end
*
case
when left(aa.dcno,2) in ('07','08','05','06') then 1
when left(aa.dcno,2) in ('01','03') then -1
when left(aa.dcno,2) in ('02','04') then 1
else 0
end
) as commission_amt,
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
0 as discount_amt,
0 as admin_fee_amt,
case
    when bb.sesiretro = 'SESI' then 0
    else 0.0002
end as pd,
case
    when bb.sesiretro = 'SESI' then 0
    else 0.45
end as lgd,
'FRA' as transition_approach_cd,
0 as transition_cumulative_oci_amt,
max(qq.ydc) as year_defisit_clause,
max(qq.management_expenses) as management_pct,
max(qq.profit_commission_calc) as profit_comm_pct,
0 as re_profit_comm_amt,
null::date re_profit_comm_dt,
aa.category as source_data,
'INITIAL' as category_data,
trim(aa.dcno) dcno,
trim(aa.dcinduk) dcinduk,
trim(aa.dcstatus) dcstatus,
aa.dcnofile,
aa.dcnoretro,
aa.dcjenis,
case
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
    else coalesce(gg.kdcob, aa.kdcob)
end as kdcob,
coalesce(
    uu.insurance_contract_id_2,
    concat_ws(
        '|',
        to_char((row_number      ()      over(partition      by
date_part('YEAR',  '${reporting_dt}':date) order by aa.dcperiodstart asc)),
'fm00000'), -- Running Number
        to_char(date_part('MONTH',
`${reporting_dt}':date), 'fm00'), -- Month
        ff.kode_lokasi_afiliasi, -- Lokasi & Afiliasi
    case
        when coalesce(gg.kdcob, aa.kdcob) = 'HE' then
'EN'
        when left(aa.dcno,2) in ('05','06') then
coalesce(gg.kdcob, aa.kdcob, 'LF')
        else coalesce(gg.kdcob, aa.kdcob)
    end, -- COB
    coalesce(kk.sub_cob_id,oo.sub_cob_id,'00'), -- SUB COB
        to_char(date_part('YEAR',coalesce(aa.dcperiodstart,'1960-01-01':date)),
'fm0000'), -- Cohort
        ff.jenis, -- Source of Business
        coalesce(ee1.idinsured,ee2.idinsured), -- Insurred
        to_char(ff.kode_ifrs::int,'fm000'), -- Ceding
    case
        when bb.sesiretro = 'SESI' then 'S'
        when bb.sesiretro = 'RETRO' then 'R'
        else '-'
    end, -- Insurance type
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        rr.kob_cd, -- KOB
        dd.coa_prefix -- Currency
    )
) as insurance_contract_id_2,
aa.dcdateloss,
(
    case
        when gg.dcno is not null then
sum(coalesce(gg.ddcclaim,0))
            else sum(coalesce(aa.dcclaim,0))
        end
    *
    case
        when left(aa.dcno,2) in ('07','08','05','06') then 1
        when left(aa.dcno,2) in ('01','03') then -1
        when left(aa.dcno,2) in ('02','04') then 1
        else 0
    end
) as dcclaim,
(
    case
        when gg.dcno is not null then
sum(coalesce(gg.ddsalvage,0))
            else sum(coalesce(aa.dcsalvage,0))
        end
    *
    case
        when left(aa.dcno,2) in ('07','08','05','06') then 1
        when left(aa.dcno,2) in ('01','03') then 1
        when left(aa.dcno,2) in ('02','04') then -1
        else 0
    end
) as dcsalvage,
(
    case
        when gg.dcno is not null then
sum(coalesce(gg.ddcbebanelaim,0))
            else sum(coalesce(aa.dcbebanelaim,0))
        end
    *
    case
        when left(aa.dcno,2) in ('07','08','05','06') then 1
        when left(aa.dcno,2) in ('01','03') then -1
        when left(aa.dcno,2) in ('02','04') then 1
        else 0
    end
) as dcbebanelaim,
aa.subcob,
aa.dcnold,
aa.dctanggal
from all_initial aa
left join staging.ref_trxcode bb ON aa.dctrxcode = bb.code
left join report.ref_currency dd ON aa.dccurr = dd.code
left join (select * from staging.ref_insured where left(dcinsured,1) != '9')
ee1 on aa.dcinsured = ee1.dcinsured
left join (select * from staging.ref_insured where left(dcinsured,1) = '9')
ee2 on aa.dcinsuredname = ee2.dcinsuredname
left join staging.ref_ceding ff on aa.cedkd::int = ff.cedkd::int
left join staging.tbl_detaildncn gg on aa.dcno = gg.dcno
left join staging.ref_sub_cob kk on aa.subcob = kk.sub_cob_name and
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

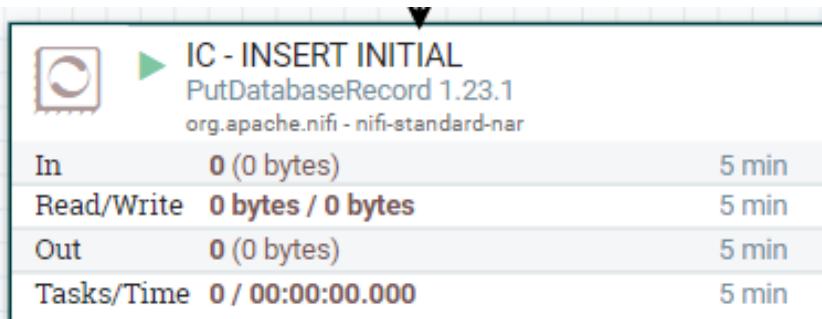


23 Februari 2024

```
case
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
        else coalesce(gg.kdcob, aa.kdcob)
    end = kk.kdcob
left join link_sesi_retro ll on aa.dcno = ll.dcno_sesi and aa.dcnofile =
ll.dcnofile and aa.kdcob = ll.kdcob and aa.subcob = ll.subcob and aa.dccurr =
ll.dccurr
left join (select kdcob, min(sub_cob_id) as sub_cob_id from
staging.ref_sub_cob group by kdcob) oo on
case
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
        else coalesce(gg.kdcob, aa.kdcob)
    end = oo.kdcob
left join staging.tbl_jurnal pp on aa.dcno = pp.jnnonota
left join (
    select
        (cob_item::json->0->'written_share'-'ydc')::text::float8 as
ydc,
        (cob_item::json->0->'written_share'-
>'management_expenses')::text::float8 management_expenses,
        (cob_item::json->0->'written_share'-
>'profit_commission_calc')::text::float8 profit_commission_calc,
        replace((bordereauxaccounts::json->'accounts')::text,'','','')
periode_pelaporan,
        treaty_number
    from staging.treaties
) qq on aa.dcnofile = qq.treaty_number
left join dcno_kob rr on aa.dcno = rr.dcno and
case
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
        else coalesce(gg.kdcob, aa.kdcob)
    end = rr.kdcob
left join (
    select dcno, count(*) n_installment from staging.tbl_installdncn ti
group by dcno
) ss on aa.dcno = ss.dcno
left join staging.ref_pattern_product_id tt on
    ss.n_installment = tt.installment and
    coalesce(aa.dcwp, DATE_PART('day', dcdue date)::timestamp) -
dcperiodestart::timestamp,0) >= tt.wpc_min and
    coalesce(aa.dcwp, DATE_PART('day', dcdue date)::timestamp) -
dcperiodestart::timestamp,0) <= tt.wpc_max
left join report.master_icid uu on aa.dcno = uu.dcno and
case
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
        else coalesce(gg.kdcob, aa.kdcob)
    end = uu.kdcob
where 1=1
    and pp.jnnstsposting = '1'
    and aa.dcstsjurnal = '4'
group by
```

```
aa.dcno,aa.dcperiodestart,aa.dendorsement_type,aa.dctanggal,aa.dcperiodeend
,aa.dccurr,aa.category,
aa.dcindeuk,aa.dcstatus,aa.dcnofile,aa.dnotaretro,aa.dcjenis,aa.dcdateloss,a
a.subcob,aa.dcnold,
bb.sesiretro,bb.name,
dd.coa_prefix,
ee1.idinsured,
ee2.idinsured,
ff.kode_lokasi_afiliasi,ff.jenis,ff.kode_ifrs,
gg.ddckot,gg.dcno,
kk.sub_cob_id,
ll.dcnosesi,
oo.sub_cob_id,
qq.periode_pelaporan,
rr.kob_cd,rr.kob_id,
ss.n_installment,
tt.pattern,
uu.insurance_contract_id,uu.insurance_contract_id_2,
case
when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
else coalesce(gg.kdcob, aa.kdcob)
end
)
select * from generate_icid_initial
```

- PutDatabaseRecord



Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel insurance_contract di database PostgreSQL DWH.

- c. Get Endorsement Data

Pada proses ini akan menghasilkan data endorsement dari tbl_dncn untuk kemudian disimpan kedalam tabel insurance_contract.

Processor yang digunakan untuk step adalah sebagai berikut:

- ExecuteSQL



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

IC - GET ENDORSEMENT BLIPSOPS

ExecuteSQL 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengeksekusi perintah SQL untuk mendapatkan data endorsement dari tbl_dncn dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

Berikut query yang digunakan untuk menghasilkan data initial:

```
with set_cancelation as (
    select
        a.dcno, a.dcnoold, a.cedkd, a.kdcob, a.dcmulticob, a.dctipedok,
        a.dcbisnisunit, a.dctrxcode, a.dckob, a.dctanggal,
        a.dcnofile, a.dcpolicyno, a.dlakindofpayments, a.dcjenis,
        a.dcasuransi, a.dccurr, a.dcgrosspremium, a.dccommission,
        a.dcintfee, a.dcbrokerage, a.dcclaim, a.dcbebanclaim,
        a.dcpocommission, a.dcovrcommission, a.dcfleetdiskon, a.dcshare,
        a.dcpremainot, a.dctax, a.dcsalvage, a.dcnet, a.dcduedate, a.dcwpc,
        a.dcpriodestart, a.dcpriodeend, a.dcinuk, a.dciinsured,
        UPPER(TRIM(a.dciinsuredname)) as dcinsuredname, a.dcnola,
        a.dcdateloss, a.dcuyear, a.dcsoa, a.dcketerangan,
        a.dcconfreff, a.dcremarkbind, a.dcrislip, a.dcourref,
        a.dcreinstatement, a.dcauth, a.dcjnkode, a.dcnetygditerima,
        a.dcnetygdibayar, a.dcnnetttotalpem, a.dcneterimig, a.dcnbalance,
        a.dctsmapping, a.dcnatemapping, a.dcsqllmapping,
        a.dctsjsurnal, a.dcpriodesoa, a.dcpriodsoquarter, a.dcpriodsoath,
        a.dctsvoxpro, a.dcsuninsured, a.dcgppremium,
        a.dcpcentpremi, a.dcpcentcomm, a.create_user, a.create_date,
        a.update_user, a.update_date, a.create_user_akseptasi,
        a.dcfasttrack, a."source", a.dcuunit, a.dcretro, a.subcob, a.dchp,
        a.qq_note, a.policy_holder, a.policy_holder_code,
        a.dcpriodestartins, a.dcpriodeendins, a.dcnnotaretro,
        'Cancellation' as dcstatus,
        a.dcvat, a.dcnoinvoice, a.dcnokontrak, a.dciendorsement_type,
        a.dcnameofrisk, a.dcpolicy_type, a.dwh_created_date
    from staging.tbl_dncn a
    left join staging.tbl_dncn b on trim(SPLIT_PART(a.dcinuk, '|', 1)) = b.dcno
    where 1=1
        and a.dcinuk is not null and a.dcinuk != ''
        and (a.dcstatus is null or a.dcstatus = '')
        and a.dcnofile = b.dcnofile
        and (
            a.dcgrosspremium
            *
            case
                when left(a.dcno,2) in ('07','08','05','06')
                when left(a.dcno,2) in ('01','03') then 1
                when left(a.dcno,2) in ('02','04') then -1
                else 0
            end
            +
        then 1
            when left(a.dcno,2) in ('01','03') then 1
            when left(a.dcno,2) in ('02','04') then -1
            else 0
        end
        +
)
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
b.dcgrosspremium
*
case
    when left(b.dcno,2)  in ('07','08','05','06')
then 1
    when left(b.dcno,2) in ('01','03') then 1
    when left(b.dcno,2) in ('02','04') then -1
    else 0
end
) = 0
),
tbl_dncn_all as (
    select
        a.dcno, a.dcnoold, a.cedkd, a.kdcob, a.dcmulticob, a.dctipedok,
a.dcbisnisunit, a.dctrxcode, a.dckob,
        a.dctanggal, a.dcnofile, a.dcpolicyno, a.dlakindofpayments, a.dcjenis,
a.dcasuransi, a.dccurr,
        a.dcgrosspremium, a.dccommission, a.dciintfee, a.dcbrokerage,
a.dccclaim, a.dcbebanclaim, a.dcpocommission,
        a.dcovrcommision, a.dcfleetdiskon, a.dcsshare, a.dcpremainot, a.dctax,
a.dcsalvage, a.dcnet, a.dcdue date,
        a.dcwpc, a.dcpriodestart, a.dcpriodeend, a.dciinduk, a.dciinsured,
UPPER(TRIM(a.dciinsuredname)) as dcinsuredname,
        a.dcnoda, a.dcdateloss, a.dcuwyear, a.dcsoa, a.dcketerangan,
a.dcconfreff, a.dcremarkbind, a.dcrislip, a.dcourref,
        a.dcreinstatement, a.dcauth, a.dcjnkode, a.dcnetygditerima,
a.dcnetygdibayar, a.dcnnetttotalpem, a.dcneterimig,
        a.dcnetsbalance, a.dcnstsmapping, a.dcnote mapping, a.dcsq1mapping,
a.dcnstsjurnal, a.dcpriodesoa,
        a.dcpriodsoquarter, a.dcpriodsoath, a.dcnstsvoxpro, a.dcsuninsured,
a.dcgrr premium, a.dcpcentpremi,
        a.dcpcentcomm, a.create_user, a.create_date, a.update_user,
a.update_date, a.create_user_akseptasi,
        a.dcfasttrack, a."source", a.dcuunit, a.dcretro, a.subcob, a.dchp,
a.qq_note, a.policy_holder,
        a.policy_holder_code, a.dcpriodestartins, a.dcpriodeendins,
a.dcnnotaretro, a.dcnstatus, a.dcvat,
        a.dcnoinvoice, a.dcnokontrak, a.dendorsement_type, a.dcnameofrisk,
a.dcpolicy_type, a.dwh_created_date
    from staging.tbl_dncn a
    left join set_cancelation b on a.dcno = b.dcno
    where b.dcno is null
    union all
    select * from set_cancelation
),
tbl_dncn_category_all as (
    select * ,
        case when dcno like '%-U%' then 'WebUpload'
            when dcno like '%-C%' and dcno not like '%-U%' then 'WebCredit'
            else 'BlipsOps'
        end as category
    from tbl_dncn_all a
    where 1=1
),
tbl_dncn_category as (
    select * from tbl_dncn_category_all a
    where 1=1
        and left(dcno,2) not in ('03','04','07','08') -- claim (03,04) & soa
treaties(07,08) di exclude
        and (

```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
(dcperiodestart      <=      '${year_reporting_dt}-01-01'      and
dcperiodeend    >=  '${year_reporting_dt}-01-01')
      or
      (dcperiodestart      >      '${year_reporting_dt}-01-01'      and
dcperiodestart <=  '${reporting_dt}' and dcperiodeend >=  '${reporting_dt}'
)
),
initial_BlipsOps as (
    select * from tbl_dncn_category
    where category = 'BlipsOps' and (
        trim(dcstatus) in ('New', 'Replacement')
        or trim(dcstatus) = ''
        or trim(dcstatus) is null
        or trim(dcinduk) is null
        or trim(dcinduk) = ''
        or (trim(dcinduk) like '%|%' and trim(SPLIT_PART(dcinduk, '|', 1)) is
null)
        or (trim(dcinduk) like '%|%' and trim(SPLIT_PART(dcinduk, '|', 1)) =
'')
    )
),
initial_WebUpload as (
    select * from tbl_dncn_category
    where category = 'WebUpload' and dcnofile not like '%.E' and dcnofile not like
'%.CL'
),
endorsement_BlipsOps as (
    select a.* from tbl_dncn_category a
    left join initial_BlipsOps b on a.dcno = b.dcno
    where a.category = 'BlipsOps' and b.dcno is null
),
endorsement_WebUpload as (
    select a.* from tbl_dncn_category a
    left join initial_WebUpload b on a.dcno = b.dcno
    where a.category = 'WebUpload' and b.dcno is null
),
endorsement_from_initial_BlipsOps1 as (
    select distinct b.dcno from initial_BlipsOps a
    join tbl_dncn_category_all b on a.dcno = trim(SPLIT_PART(b.dcinduk, '|', 1))
),
endorsement_from_initial_BlipsOps2 as (
    select * from endorsement_from_initial_BlipsOps1
    union all
    select distinct b.dcno from endorsement_from_initial_BlipsOps1 a
    join tbl_dncn_category_all b on a.dcno = trim(SPLIT_PART(b.dcinduk, '|', 1))
),
endorsement_from_initial_BlipsOps3 as (
    select * from endorsement_from_initial_BlipsOps2
    union all
    select distinct b.dcno from endorsement_from_initial_BlipsOps2 a
    join tbl_dncn_category_all b on a.dcno = trim(SPLIT_PART(b.dcinduk, '|', 1))
),
endorsement_from_initial_BlipsOps4 as (
    select * from endorsement_from_initial_BlipsOps3
    union all
    select distinct b.dcno from endorsement_from_initial_BlipsOps3 a
    join tbl_dncn_category_all b on a.dcno = trim(SPLIT_PART(b.dcinduk, '|', 1))
),
endorsement_from_initial_BlipsOps5 as (
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
select * from endorsement_from_initial_BlipsOps4
union all
select distinct b.dcno from endorsement_from_initial_BlipsOps4 a
join tbl_dncn_category_all b on a.dcno = trim(SPLIT_PART(b.dcinduk, '|', 1))
),
endorsement_from_initial_BlipsOps6 as (
    select * from endorsement_from_initial_BlipsOps5
    union all
    select distinct b.dcno from endorsement_from_initial_BlipsOps5 a
    join tbl_dncn_category_all b on a.dcno = trim(SPLIT_PART(b.dcinduk, '|', 1))
),
endorsement_from_initial_BlipsOps7 as (
    select * from endorsement_from_initial_BlipsOps6
    union all
    select distinct b.dcno from endorsement_from_initial_BlipsOps6 a
    join tbl_dncn_category_all b on a.dcno = trim(SPLIT_PART(b.dcinduk, '|', 1))
),
data_admnf2013 as (
    select nofileinduk, orc, min(nodebit) nodebit, count(*) cnt
    from staging.admnf2013
    group by nofileinduk, orc
),
data_admnf2013_1 as (
    select nofileinduk, orc, nodebit from data_admnf2013 where cnt = 1
),
data_admnf2013_2 as (
    select
        nofileinduk, orc,
        replace(nodebit,concat('.','.',split_part(nodebit,'.',4),'.'),'.00.')
nodebit
    from data_admnf2013 where cnt > 1
),
data_admnf2013_union as (
    select * from data_admnf2013_1
    union all
    select * from data_admnf2013_2
),
endorsement_WebUpload_find_dcinduk as (
    select
        c.assasment,
        c.nofileinduk,
        b.nodebit,
        a.dcno,
        a.dccurr
    from endorsement_WebUpload a
    left join (
        select distinct nodebit,kodecdn,nofileinduk from staging.admnf2013
    ) b on a.dcnold = b.nodebit and case when a.dcjenis = 1 then 'DN' else 'CN'
end = b.kodecdn
    left join (
        select distinct nofileinduk,assasment from staging.admnf2000
    ) c on b.nofileinduk = c.nofileinduk
),
endorsement_WebUpload_ketemu_dcinduk as (
    select distinct
        d.dcno as dcinduk,
        a.dcno,
        a.assasment_nofile3
    from (
        select * from (

```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        select dcno, dccurr,
               split_part(assasment_nofile3,'.',4) as
assasment_nofile3_year,
               assasment_nofile3,
               row_number() over (partition by dcno, dccurr order by
split_part(assasment_nofile3,'.',4) desc) as nrow
        from (
               select
                     dcno,
                    dccurr,
                     (regexp_matches(encode(assasment,'escape'),
'([^\s\.]+\.[^s\.]+\.[^s\.]+\.[0-9]{4}\.[^\s\.]+)', 'g'))[1] as
assasment_nofile3
                     from endorsement_WebUpload_find_dcinduk
               ) x
        ) y where nrow = 1
      ) a
      left join (
            select distinct nofile3,nofileinduk from staging.admnf2000
      ) b on a.assasment_nofile3 = b.nofile3
      left join data_admnf2013_union c on b.nofileinduk = c.nofileinduk and a.dccurr
= c.orc
      left join (
            select distinct dcnoold,dcnofile,dcno from staging.tbl_dncn
      ) d on c.nodebit = d.dcnoold and a.assasment_nofile3 = d.dcnofile
),
      endorsement_WebUpload_get_dcinduk as (
      select
            a.dcno, a.dcnoold, a.cedkd, a.kdcob, a.dcmulticob, a.dctipedok,
a.dcbisnisunit, a.dctrxcode, a.dckob, a.dctanggal,
            a.dcnofile, a.dcpolicyno, a.dlakindofpayments, a.dcjenis,
a.dcasuransi, a.dccurr, a.dcgrosspremium, a.dccommission,
            a.dciintfee, a.dcbrokerage, a.dccclaim, a.dcbebanclaim,
a.dcpocommission, a.dcovrcommision, a.dcfleetdiskon, a.dcsshare,
            a.dcpreminot, a.dctax, a.dcsalvage, a.dcnet, a.dcdue date, a.dcwpc,
a.dcpriodestart, a.dcpriodeend,
            b.dcinduk,
            a.dci nsured, a.dci nsuredname, a.dcnodla, a.dcdateloss, a.dcuwyear,
a.dcs oa, a.dcketerangan, a.dcconfreff,
            a.dcremarkbind, a.dcrislip, a.dcourref, a.dcreinstatement, a.dcauth,
a.dcjnkode, a.dcnetygditerima, a.dcnetygdibayar,
            a.dcnnetttotalpem, a.dcneterimig, a.dcnbalance, a.dcts mapping,
a.dcnote mapping, a.dcsqlmapping, a.dcts jurnal,
            a.dcpriodessoa, a.dcpriodsoaquarter, a.dcpriodsoath, a.dcts voxpro,
a.dcs uminsured, a.dcg r premium, a.dcpri centpremi,
            a.dcpri centcomm, a.create user, a.create date, a.update user,
a.update date, a.create user akseptasi, a.dcfasttrack,
            a."source", a.dcu nity, a.dcretro, a.subcob, a.dchp, a.qq note,
a.policy holder, a.policy holder code, a.dcpriodestartins,
            a.dcpriodeendins, a.dcnote retro, a.dccstatus, a.dcvat, a.dcnoinvoice,
a.dcnokontrak, a.dendorsement type, a.dcnameofrisk,
            a.dcpolicy type, a.dwh created date, a.category
      from endorsement_WebUpload a
      left join endorsement_WebUpload_ketemu_dcinduk b on a.dcno = b.dcno
),
      all_endorsement as (
            select * from endorsement_BlapsOps
            union all
            select * from endorsement_WebUpload_get_dcinduk
      ),
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
distinct_dcno_tbl_dncn as (
    select distinct dcno, dcinduk from staging.tbl_dncn
),
distinct_dcno_master_icid as (
    select distinct dcno from report.master_icid
),
distinct_dcno1 as (
    select distinct dcno from all_endorsement
),
get_induk1 as (
    select distinct b.dcno, b.dcinduk, mi.dcno as mi_dcno from all_endorsement a
    left join distinct_dcno_tbl_dncn b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno
    left join distinct_dcno_master_icid mi on b.dcno = mi.dcno
),
distinct_dcno2 as (
    select * from distinct_dcno1
    union all
    select distinct dcno from get_induk1 where mi_dcno is not null
),
get_induk2 as (
    select distinct b.dcno, b.dcinduk, mi.dcno as mi_dcno from get_induk1 a
    left join distinct_dcno_tbl_dncn b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno
    left join distinct_dcno_master_icid mi on b.dcno = mi.dcno
    where mi_dcno is null
),
distinct_dcno3 as (
    select * from distinct_dcno2
    union all
    select distinct dcno from get_induk2 where mi_dcno is not null
),
get_induk3 as (
    select distinct b.dcno, b.dcinduk, mi.dcno as mi_dcno from get_induk2 a
    left join distinct_dcno_tbl_dncn b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno
    left join distinct_dcno_master_icid mi on b.dcno = mi.dcno
    where mi_dcno is null
),
distinct_dcno4 as (
    select * from distinct_dcno3
    union all
    select distinct dcno from get_induk3 where mi_dcno is not null
),
get_induk4 as (
    select distinct b.dcno, b.dcinduk, mi.dcno as mi_dcno from get_induk3 a
    left join distinct_dcno_tbl_dncn b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno
    left join distinct_dcno_master_icid mi on b.dcno = mi.dcno
    where mi_dcno is null
),
distinct_dcno5 as (
    select * from distinct_dcno4
    union all
    select distinct dcno from get_induk4 where mi_dcno is not null
),
get_induk5 as (
    select distinct b.dcno, b.dcinduk, mi.dcno as mi_dcno from get_induk4 a
    left join distinct_dcno_tbl_dncn b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
        left join distinct_dcno_master_icid mi on b.dcno = mi.dcno
        where mi_dcno is null
),
distinct_dcno6 as (
    select * from distinct_dcno5
    union all
    select distinct dcno from get_induk5 where mi_dcno is not null
),
get_induk6 as (
    select distinct b.dcno, b.dcinduk, mi.dcno as mi_dcno from get_induk5 a
    left join distinct_dcno_tbl_dncn b on trim(SPLIT_PART(a.dcinduk, '|', 1)) =
b.dcno
    left join distinct_dcno_master_icid mi on b.dcno = mi.dcno
    where mi_dcno is null
),
distinct_dcno7 as (
    select * from distinct_dcno6
    union all
    select distinct dcno from get_induk6 where mi_dcno is not null
),
get_induk7 as (
    select distinct b.dcno, b.dcinduk, mi.dcno as mi_dcno from get_induk5 a
    left join distinct_dcno_tbl_dncn b on trim(SPLIT_PART(a.dcinduk, '|', 1)) =
b.dcno
    left join distinct_dcno_master_icid mi on b.dcno = mi.dcno
    where mi_dcno is null
),
distinct_dcno8 as (
    select * from distinct_dcno7
    union all
    select distinct dcno from get_induk7 where mi_dcno is not null
    union all
    select distinct dcno from endorsement_from_initial_BlipsOps7
),
distinct_dcno8_join_tbl_dncn as (
    select a.* from (
        select a.* from tbl_dncn_category_all a
        join (select distinct dcno from distinct_dcno8) b on a.dcno = b.dcno
    ) a
    left join (select distinct dcno from report.${insurance_contract_step1}) b on
a.dcno = b.dcno
    where 1=1 and b.dcno is null
),
tbl_dncn_category_per_dcno as (
    select a.dcno, dcnofile, subcob,
    case
        when coalesce(b.kdcob, a.kdcob) = 'HE' then 'EN'
        when left(a.dcno,2) in ('05','06') then coalesce(b.kdcob,
a.kdcob, 'LF')
            else coalesce(b.kdcob, a.kdcob)
        end as kdcob,
        a.dccurr,
        case
            when b.dcno is not null then sum(coalesce(b.ddcpremium,0))
            else sum(coalesce(a.dcgrosspremium,0))
        end
    *
    case
        when left(a.dcno,2) in ('07','08','05','06') then 1
        when left(a.dcno,2) in ('01','03') then 1
    end
)
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
        when left(a.dcno,2) in ('02','04') then -1
        else 0
    end    as prem_amt
from tbl_dncn_category a
left join staging.tbl_detaildncn b on a.dcno = b.dcno
group by a.dcno, b.dcno, a.kdcob, b.kdcob, a.dcnofile, a.subcob, a.dccurr
),
link_sesi_retro_dwh as (
    select
        a.dcno as dcno_sesi,
        a.dcnofile, a.kdcob, a.subcob, a.dccurr,
        string_agg(b.dcno,'|') dcno_retro,
        max(a.prem_amt) as prem_amt_sesi,
        sum(b.prem_amt) as prem_amt_retro
    from (
        select * from tbl_dncn_category_per_dcno a where a.dcno like '%K11%'
    ) a
    join (
        select * from tbl_dncn_category_per_dcno a where a.dcno like '%K51%'
    ) b on a.dcnofile = b.dcnofile and a.kdcob = b.kdcob and a.subcob = b.subcob
and a.dccurr = b.dccurr
        group by a.dcno, a.kdcob, b.kdcob, a.subcob, a.dccurr, a.dcnofile
),
link_sesi_retro_manual as (
    select a.*, b.prem_amt as prem_amt_sesi, c.prem_amt as prem_amt_retro from (
        select distinct dcno_sesi, dcno, dcnofile, kdcob, subcob, dccurr from report.dcno_retro_not_connected
            where dcno_sesi is null or dcno_sesi != ''
    ) a
    left join tbl_dncn_category_per_dcno b on a.dcno_sesi = b.dcno
    left join tbl_dncn_category_per_dcno c on a.dcno = c.dcno
),
link_sesi_retro as (
    select distinct
        coalesce(a.dcno_sesi,b.dcno_sesi) as dcno_sesi,
        coalesce(a.dcnofile,b.dcnofile) as dcnofile,
        coalesce(a.kdcob,b.kdcob) as kdcob,
        coalesce(a.subcob,b.subcob) as subcob,
        coalesce(a.dccurr,b.dccurr) as dccurr,
        coalesce(a.dcno_retro,b.dcno) as dcno_retro,
        coalesce(a.prem_amt_sesi,b.prem_amt_sesi) as prem_amt_sesi,
        coalesce(a.prem_amt_retro,b.prem_amt_retro) as prem_amt_retro
    from link_sesi_retro_dwh a
        full outer join link_sesi_retro_manual b on a.dcno_sesi = b.dcno_sesi and
a.dcno_retro = b.dcno
),
dcno_kob1 as (
    select distinct
        aa.dcno,
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
                else coalesce(gg.kdcob, aa.kdcob)
        end as kdcob,
        case
            when aa.dctrxcode in ('11','51') then '5'
            when aa.dctrxcode = '22' then
                case
                    when ii.jenis = 'SOA Life' then '4'
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
else
    case
        when nn.jenis = 'SOA Treaty QS'
        when nn.jenis = 'SOA Treaty SPL'
        when nn.jenis = 'SOA Treaty FACOB'
        else '4'
    end
end kob_cd,
case
    when aa.dctrxcode in ('11','51') then 'FC'
    when aa.dctrxcode = '22' then
        case
            when ii.jenis = 'SOA Life' then 'TN'
            else
                case
                    when nn.jenis = 'SOA Treaty QS'
                    when nn.jenis = 'SOA Treaty SPL'
                    when nn.jenis = 'SOA Treaty FACOB'
                    else 'TN'
                end
            end
        end
    else 'TN'
end as kob_id
from distinct_dcno8_join_tbl_dncn aa
left join staging.tbl_detaildncn gg on aa.dcno = gg.dcno
left join staging.ref_tipedok ii on aa.dctipedok = ii.code
left join staging.ref_tipedok nn on gg.ddckot = nn.code
),
initial_BlipsOps2 as (
    select * from distinct_dcno8_join_tbl_dncn
    where category = 'BlipsOps' and (
        trim(dcstatus) in ('New', 'Replacement')
        or trim(dcstatus) = ''
        or trim(dcstatus) is null
        or trim(dcinduk) is null
        or trim(dcinduk) = ''
        or (trim(dcinduk) like '%|%' and trim(SPLIT_PART(dcinduk, '|', 1)) is
null)
        or (trim(dcinduk) like '%|%' and trim(SPLIT_PART(dcinduk, '|', 1)) =
'')
    )
),
initial_WebUpload2 as (
    select * from distinct_dcno8_join_tbl_dncn
    where category = 'WebUpload' and dcnofile not like '%.E' and dcnofile not like
'%.CL'
),
initial_WebCredit2 as (
    select * from distinct_dcno8_join_tbl_dncn
    where category = 'WebCredit'
),
all_initial2 as (
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
select * from initial_BlipsOps2
union all
select * from initial_WebUpload2
union all
select * from initial_WebCredit2
),
endorsement_BlipsOps2 as (
    select a.* from distinct_dcno8_join_tbl_dncn a
    left join initial_BlipsOps2 b on a.dcno = b.dcno
    where a.category = 'BlipsOps' and b.dcno is null
),
endorsement_WebUpload2 as (
    select a.* from distinct_dcno8_join_tbl_dncn a
    left join initial_WebUpload2 b on a.dcno = b.dcno
    where a.category = 'WebUpload' and b.dcno is null
),
endorsement_WebUpload_find_dcinduk2 as (
    select
        c.assasment,
        c.nofileinduk,
        b.nodebit,
        a.dcno,
        a.dccurr
    from endorsement_WebUpload2 a
    left join (
        select distinct nodebit,kodecdn,nofileinduk from staging.admnf2013
    ) b on a.dcnoold = b.nodebit and case when a.dcjenis = 1 then 'DN' else 'CN'
end = b.kodecdn
    left join (
        select distinct nofileinduk,assasment from staging.admnf2000
    ) c on b.nofileinduk = c.nofileinduk
),
endorsement_WebUpload_ketemu_dcinduk2 as (
    select distinct
        d.dcno as dcinduk,
        a.dcno,
        a.assasment_nofile3
    from (
        select * from (
            select dcno, dccurr,
                split_part(assasment_nofile3,'.',4) as assasment_nofile3_year,
                assasment_nofile3,
                row_number() over (partition by dcno, dccurr order by
                split_part(assasment_nofile3,'.',4) desc) as nrow
            from (
                select
                    dcno,
                    dccurr,
                    (regexp_matches(encode(assasment,'escape'),
                    '([^\s\.]+'\.[^\s\.]+'\.[^\s\.]+'\.[0-9]{4}\.[^\s\.]+'),'g'))[1] as assasment_nofile3
                from endorsement_WebUpload_find_dcinduk2
            ) x
        ) y where nrow = 1
    ) a
    left join (
        select distinct nofile3,nofileinduk from staging.admnf2000
    ) b on a.assasment_nofile3 = b.nofile3
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
    left join data_admnf2013_union c on b.nofileinduk = c.nofileinduk and a.dccurr
= c.orc
    left join (
        select distinct dcnoold,dcnofile,dcno from staging.tbl_dncn
    ) d on c.nodebit = d.dcnoold and a.assasment_nofile3 = d.dcnofile
),
endorsement_WebUpload_get_dcinduk2 as (
    select
        a.dcno, a.dcnooldd, a.cedkd, a.kdcob, a.dcmulticob, a.dctipedok,
a.dcbisnisunit, a.dctrxcode, a.dckob, a.dctanggal,
        a.dcnofile, a.dcpolicyno, a.dlakindofpayments, a.dcjenis,
a.dcasuransi, a.dccurr, a.dcgrosspremium, a.dccommission,
        a.dciintfee, a.dcbrokerage, a.dccclaim, a.dcbebanclaim,
a.dcpocommission, a.dcovrcommision, a.dcfleetdiskon, a.dcshare,
        a.dcpremainot, a.dctax, a.dcsalvage, a.dcnet, a.dcdue date, a.dcwpc,
a.dcpriodestart, a.dcpriodeend,
        b.dcinduk,
        a.dciinsured, a.dciinsuredname, a.dcnodla, a.dcdateloss, a.dcuwyyear,
a.dcsoa, a.dcketerangan, a.dcconfreff,
        a.dcremarkbind, a.dcrislip, a.dcourref, a.dcreinstatement, a.dcauth,
a.dcjnkode, a.dcnetygdirima, a.dcnetygdibayar,
        a.dcnnetttotalpem, a.dcneterimig, a.dcnbalance, a.dctsmapping,
a.dcnnotemapping, a.dcsqllmapping, a.dctsjsurnal,
        a.dcpriodessoa, a.dcpriodsoquarter, a.dcpriodsoath, a.dctsvoxpro,
a.dcsuinsured, a.dcgppremium, a.dcpcentpremi,
        a.dcpcentcomm, a.create_user, a.create_date, a.update_user,
a.update_date, a.create_user_akseptasi, a.dcfasttrack,
        a."source", a.dcuunit, a.dcretro, a.subcob, a.dchp, a.qq_note,
a.policy_holder, a.policy_holder_code, a.dcpriodestartins,
        a.dcpriodeendins, a.dcnnotaretro, a.dccstatus, a.dcvat, a.dcnoinvoice,
a.dcnokontrak, a.dcdorsement_type, a.dcnnameofrisk,
        a.dcpolicy_type, a.dwh_created_date, a.category
    from endorsement_WebUpload2 a
    left join endorsement_WebUpload_ketemu_dcinduk2 b on a.dcno = b.dcno
),
all_endorsement2 as (
    select * from endorsement_BlapsOps2
    union all
    select * from endorsement_WebUpload_get_dcinduk2
),
generate_endorsement as (
    select
        '${reporting_dt}':date as reporting_dt,
        'TRE' as entity_id,
        coalesce(ich.insurance_contract_id,'-') as insurance_contract_id,
        '99' as endorsement_type_cd,
        coalesce(aa.dcpriodestart,'1960-01-01':date) as issue_dt,
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
                else coalesce(gg.kdcob, aa.kdcob)
        end as product_line_id,
        rr.kob_id,
        concat(
            coalesce(kk.sub_cob_id,oo.sub_cob_id,'00'),
            '_',
            rr.kob_id,
            '_',
            case
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
when
(
    (12 - EXTRACT(MONTH FROM dcperiodestart))
+
        (EXTRACT(MONTH FROM dcperiodeend)) +
        ((EXTRACT(YEAR FROM dcperiodeend)) -
EXTRACT(YEAR FROM dcperiodestart) - 1) * 12) +
        (case when EXTRACT(DAY FROM dcperiodeend)
- EXTRACT(DAY FROM dcperiodestart) > 0 then 1 else 0 end)
        ) = 0 then 1
    else
        (
            (12 - EXTRACT(MONTH FROM dcperiodestart))
+
                (EXTRACT(MONTH FROM dcperiodeend)) +
                ((EXTRACT(YEAR FROM dcperiodeend)) -
EXTRACT(YEAR FROM dcperiodestart) - 1) * 12) +
                (case when EXTRACT(DAY FROM dcperiodeend)
- EXTRACT(DAY FROM dcperiodestart) > 0 then 1 else 0 end)
            )
        end,'mth',
    case
        when rr.kob_id      in ('TQ','TS')      then
concat('_',upper(qq.periode_pelaporan))
        when rr.kob_id      in ('FC','TN')      then
concat('_INSTALL',ss.n_installment,'_',tt.pattern)
        else ''
    end
) as product_id,
coalesce(ff.jenis,'') as source_of_business,
to_char(ff.kode_ifrs::int,'fm000') as ceding_id,
to_char(date_part('YEAR',coalesce(aa.dcperiodestart,'1960-01-01)::date)), 'fm0000') as cohort_id,
case
    when bb.sesiretro = 'SESI' then 'N'
    when bb.sesiretro = 'RETRO' then 'Y'
    else '-'
end as ceded_flg,
case
    when bb.sesiretro = 'SESI' then 'N'
    else
        case
            when left(aa.dccno,2) in ('07','08') then
                case
                    when gg.ddckot      in
('15','16','17') then 'Y'
                    else 'N'
                end
            else 'Y'
        end
    end as reins_prop_cover_flg,
case
    when bb.sesiretro = 'SESI' then 'N'
    else
        case
            when bb."name" like '%Treaty%' then 'Y'
            else 'N'
        end
    end as reins_treaty_flg,
dctanggal as endorsement_dt,
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
coalesce(aa.dcpriodestart,'1960-01-01)::date) as begin_cov_dt,
coalesce(aa.dcpriodeend,'1960-01-01)::date) as end_cov_dt,
aa.dccurr as currency_cd,
(
    case
        when gg.dcno      is      not      null      then
            sum(coalesce(gg.ddcpremium,0))
        else sum(coalesce(aa.dcgrosspremium,0))
    end
    *
    case
        when left(aa.dcno,2) in ('07','08','05','06') then 1
        when left(aa.dcno,2) in ('01','03') then 1
        when left(aa.dcno,2) in ('02','04') then -1
        else 0
    end
) as prem_amt,
case
    when bb.sesiretro = 'RETRO' then 0
    when ll.dcno_sesi is not null then
        (
            case
                when gg.dcno      is      not      null      then
                    sum(coalesce(gg.ddcpremium,0))
                else sum(coalesce(aa.dcgrosspremium,0))
            end
            *
            case
                when         left(aa.dcno,2)      in
('07','08','05','06') then 1
                when left(aa.dcno,2) in ('01','03') then
1
                when left(aa.dcno,2) in ('02','04') then
-1
                else 0
            end
        )
        +
        sum(ll.prem_amt_retro)
    else
        case
            when gg.dcno      is      not      null      then
                sum(coalesce(gg.ddcpremium,0))
            else sum(coalesce(aa.dcgrosspremium,0))
        end
        *
        case
            when left(aa.dcno,2) in ('07','08','05','06')
then 1
            when left(aa.dcno,2) in ('01','03') then 1
            when left(aa.dcno,2) in ('02','04') then -1
            else 0
        end
    end as base_alloc,
(
    case
        when gg.dcno      is      not      null      then
            sum(coalesce(gg.ddcbrokerage,0))
        else sum(coalesce(aa.dcbrokerage,0))
    end
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
*  
case  
    when left(aa.dcno,2) in ('07','08','05','06') then 1  
    when left(aa.dcno,2) in ('01','03') then -1  
    when left(aa.dcno,2) in ('02','04') then 1  
    else 0  
end  
) as comm_broker_amt,  
(  
    case  
        when gg.dcno is not null then  
sum(coalesce(gg.ddccommission,0))  
            else sum(coalesce(aa.dccommission,0))  
        end  
    *  
    case  
        when left(aa.dcno,2) in ('07','08','05','06') then 1  
        when left(aa.dcno,2) in ('01','03') then -1  
        when left(aa.dcno,2) in ('02','04') then 1  
        else 0  
    end  
) as commission_amt,  
0 as discount_amt,  
0 as admin_fee_amt,  
case  
    when bb.sesiretro = 'SESI' then 0  
    else 0.002  
end as pd,  
case  
    when bb.sesiretro = 'SESI' then 0  
    else 0.45  
end as lgd,'FRA' as transition_approach_cd,  
0 as transition_cumulative_oci_amt,  
max(qq.ydc) as year_defisit_clause,  
max(qq.management_expenses) as management_pct,  
max(qq.profit_commission_calc) as profit_comm_pct,  
0 as re_profit_comm_amt,  
null::date re_profit_comm_dt,  
aa.category as source_data,  
'ENDORSEMENT' as category_data,  
trim(aa.dcno) dcno,  
trim(aa.dcinduk) dcinduk,  
trim(aa.dcstatus) dcstatus,  
aa.dcnofile,  
aa.dcnoretro,  
aa.dcjenis,  
case  
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'  
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,  
aa.kdcob, 'LF')  
        else coalesce(gg.kdcob, aa.kdcob)  
    end as kdcob,  
coalesce(ich.insurance_contract_id_2,'-') as insurance_contract_id_2,  
aa.dcdateloss,  
(  
    case  
        when gg.dcno is not null then  
sum(coalesce(gg.ddcclaim,0))  
            else sum(coalesce(aa.dcclaim,0))  
    end
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
*  
case  
    when left(aa.dcno,2) in ('07','08','05','06') then 1  
    when left(aa.dcno,2) in ('01','03') then -1  
    when left(aa.dcno,2) in ('02','04') then 1  
    else 0  
end  
) as dcclaim,  
(  
    case  
        when gg.dcno is not null then  
sum(coalesce(gg.ddsalvage,0))  
            else sum(coalesce(aa.dcsalvage,0))  
end  
*  
case  
    when left(aa.dcno,2) in ('07','08','05','06') then 1  
    when left(aa.dcno,2) in ('01','03') then 1  
    when left(aa.dcno,2) in ('02','04') then -1  
    else 0  
end  
) as dcsalvage,  
(  
    case  
        when gg.dcno is not null then  
sum(coalesce(gg.ddcbebanclaim,0))  
            else sum(coalesce(aa.dcbebanclaim,0))  
end  
*  
case  
    when left(aa.dcno,2) in ('07','08','05','06') then 1  
    when left(aa.dcno,2) in ('01','03') then -1  
    when left(aa.dcno,2) in ('02','04') then 1  
    else 0  
end  
) as dcbebanclaim,  
ff.kode_lokasi_afiliasi,  
coalesce(kk.sub_cob_id,oo.sub_cob_id,'00') sub_cob_id,  
coalesce(ee1.idinsured,ee2.idinsured) idinsured,  
aa.dctrxcode,  
ii.jenis_jenis_dctipedok,  
nn.jenis_jenis_ddckot,  
dd.coa_prefix coa_prefix_currency,  
aa.subcob,  
rr.kob_cd,  
aa.dcnold,  
aa.dctanggal  
from all_endorsement2 aa  
left join staging.ref_trxcode bb ON aa.dctrxcode = bb.code  
left join report.ref_currency dd ON aa.dccurr = dd.code  
left join (select * from staging.ref_insured where left(dcinsured,1) != '9') ee1 on aa.dcinsured = ee1.dcinsured  
left join (select * from staging.ref_insured where left(dcinsured,1) = '9') ee2 on aa.dcinsuredname = ee2.dcinsuredname  
left join staging.ref_ceding ff on aa.cedkd::int = ff.cedkd::int  
left join staging.tbl_detaildcn gg on aa.dcno = gg.dcno  
left join staging.ref_tipedok ii on aa.dctipedok = ii.code  
left join staging.ref_sub_cob kk on aa.subcob = kk.sub_cob_name and  
case  
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end = kk.kdcob
    left join link_sesi_retro ll on aa.dcno = ll.dcno_sesi and aa.dcnofile =
ll.dcnofile and aa.kdcob = ll.kdcob and aa.subcob = ll.subcob and aa.dccurr =
ll.dccurr
    left join staging.ref_tipedok nn on gg.ddckot = nn.code
    left join (select kdcob, min(sub_cob_id) as sub_cob_id from
staging.ref_sub_cob group by kdcob) oo on
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end = oo.kdcob
    left join staging.tbl_jurnal pp on aa.dcno = pp.jnnonota
    left join (
        select distinct
            (cob_item::json->0->'written_share'-'ydc')::text::float8 as
ydc,
            (cob_item::json->0->'written_share'-
>'management_expenses')::text::float8 management_expenses,
            (cob_item::json->0->'written_share'-
>'profit_commission_calc')::text::float8 profit_commission_calc,
            replace((bordereauxaccounts::json->'accounts')::text,'','','')
periode_pelaporan,
            treaty_number
        from staging.treaties
    ) qq on aa.dcnofile = qq.treaty_number
    left join dcno_kob1 rr on aa.dcno = rr.dcno and
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end = rr.kdcob
    left join (
        select dcno, count(*) n_installment from staging.tbl_installdncn ti
group by dcno
    ) ss on aa.dcno = ss.dcno
    left join staging.ref_pattern_product_id tt on
        ss.n_installment = tt.installment and
        coalesce(aa.dcwp, DATE_PART('day', dcdue date)::timestamp) -
dcperiodestart::timestamp,0) >= tt.wpc_min and
        coalesce(aa.dcwp, DATE_PART('day', dcdue date)::timestamp) -
dcperiodestart::timestamp,0) <= tt.wpc_max
    left join (
        select
            dcno,insurance_contract_id,kdcob,insurance_contract_id_2
            from report.master_icid
    ) ich on trim(SPLIT_PART(aa.dcinduk, '|', 1)) = ich.dcno and
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end = ich.kdcob
    where 1=1
        and pp.jnnstposting = '1'
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
        and aa.dcstsjurnal = '4'
group by

    aa.dccno,aa.dccperiodestart,aa.dccendorsement_type,aa.dcctanggal,aa.dccperiodeend
    ,aa.dcccurr,aa.category,

    aa.dccinduk,aa.dccstatus,aa.dccnofile,aa.dccnotaretro,aa.dccjenis,aa.dccdateloss,a
    a.subcob,aa.dccnold,
        aa.dccrxcode,
        bb.sesiretro,bb.name,
        dd.coa_prefix,
        ee1.idinsured,
        ee2.idinsured,
        ff.kode_lokasi_afiliasi,ff.jenis,ff.kode_ifrs,
        gg.ddckot,gg.dccno,
        ii.jenis,
        kk.sub_cob_id,
        ll.dccno_sesi,
        nn.jenis,
        oo.sub_cob_id,
        qq.periode_pelaporan,
        rr.kob_cd,rr.kob_id,
        ss.n_installment,
        tt.pattern,
        ich.insurance_contract_id,ich.insurance_contract_id_2,
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dccno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end
),
generate_endorsement2 as (
    select * from generate_endorsement where insurance_contract_id is not null and
insurance_contract_id != '-'
    union all
    select
        a.reporting_dt, a.entity_id,
        b.insurance_contract_id,
        a.endorsement_type_cd, a.issue_dt, a.product_line_id, a.kob_id,
        a.product_id, a.source_of_business, a.ceding_id, a.cohort_id,
        a.ceded_flg, a.reins_prop_cover_flg, a.reins_treaty_flg,
        a.endorsement_dt, a.begin_cov_dt, a.end_cov_dt, a.currency_cd,
        a.prem_amt, a.base_alloc, a.comm_broker_amt, a.commission_amt,
        a.discount_amt, a.admin_fee_amt, a.pd, a.lgd,
        a.transition_approach_cd, a.transition_cumulative_oci_amt,
        a.year_defisit_clause,
        a.management_pct, a.profit_comm_pct, a.re_profit_comm_amt,
        a.re_profit_comm_dt, a.source_data, a.category_data, a.dccno,
        a.dccinduk, a.dccstatus, a.dccnofile, a.dccnotaretro, a.dccjenis, a.kdcob,
        b.insurance_contract_id_2,
        a.dccdateloss, a.dccclaim,
        a.dccsalvage, a.dccbebanclaim, a.kode_lokasi_afiliasi, a.sub_cob_id,
        a.idinsured, a.dccrxcode,
        a.jenis_dctipedok, a.jenis_ddckot, a.coa_prefix_currency, a.subcob,
        a.kob_cd, a.dccnold, a.dcctanggal
    from generate_endorsement a
    left join (
        select      distinct      dcno,      kdcob,      insurance_contract_id,
        insurance_contract_id_2 from generate_endorsement
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        where insurance_contract_id is not null and insurance_contract_id !=  
        '-'  
    ) b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno and a.kdcob = b.kdcob  
    where a.insurance_contract_id is null or a.insurance_contract_id = '-'  
,  
generate_endorsement3 as (  
    select * from generate_endorsement2 where insurance_contract_id is not null  
and insurance_contract_id != '-'  
    union all  
    select  
        a.reporting_dt, a.entity_id,  
        b.insurance_contract_id,  
        a.endorsement_type_cd, a.issue_dt, a.product_line_id, a.kob_id,  
        a.product_id, a.source_of_business, a.ceding_id, a.cohort_id,  
        a.ceded_flg, a.reins_prop_cover_flg, a.reins_treaty_flg,  
        a.endorsement_dt, a.begin_cov_dt, a.end_cov_dt, a.currency_cd,  
        a.prem_amt, a.base_alloc, a.comm_broker_amt, a.commission_amt,  
        a.discount_amt, a.admin_fee_amt, a.pd, a.lgd,  
        a.transition_approach_cd, a.transition_cumulative_oci_amt,  
        a.year_defisit_clause,  
        a.management_pct, a.profit_comm_pct, a.re_profit_comm_amt,  
        a.re_profit_comm_dt, a.source_data, a.category_data, a.dcno,  
        a.dcinduk, a.dcstatus, a.dcnofile, a.dcnoretro, a.dcjenis, a.kdcob,  
        b.insurance_contract_id_2,  
        a.dcdateloss, a.dccclaim,  
        a.dcsalvage, a.dcbefanclaim, a.kode_lokasi_afiliasi, a.sub_cob_id,  
a.idinsured, a.dctrxcode,  
        a.jenis_dctipedok, a.jenis_ddckot, a.coa_prefix_currency, a.subcob,  
a.kob_cd, a.dcnoold, a.dctanggal  
    from generate_endorsement2 a  
    left join (  
        select distinct dcno, kdcob, insurance_contract_id,  
insurance_contract_id_2 from generate_endorsement2  
        where insurance_contract_id is not null and insurance_contract_id !=  
        '-'  
    ) b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno and a.kdcob = b.kdcob  
    where a.insurance_contract_id is null or a.insurance_contract_id = '-'  
,  
generate_endorsement4 as (  
    select * from generate_endorsement3 where insurance_contract_id is not null  
and insurance_contract_id != '-'  
    union all  
    select  
        a.reporting_dt, a.entity_id,  
        b.insurance_contract_id,  
        a.endorsement_type_cd, a.issue_dt, a.product_line_id, a.kob_id,  
        a.product_id, a.source_of_business, a.ceding_id, a.cohort_id,  
        a.ceded_flg, a.reins_prop_cover_flg, a.reins_treaty_flg,  
        a.endorsement_dt, a.begin_cov_dt, a.end_cov_dt, a.currency_cd,  
        a.prem_amt, a.base_alloc, a.comm_broker_amt, a.commission_amt,  
        a.discount_amt, a.admin_fee_amt, a.pd, a.lgd,  
        a.transition_approach_cd, a.transition_cumulative_oci_amt,  
        a.year_defisit_clause,  
        a.management_pct, a.profit_comm_pct, a.re_profit_comm_amt,  
        a.re_profit_comm_dt, a.source_data, a.category_data, a.dcno,  
        a.dcinduk, a.dcstatus, a.dcnofile, a.dcnoretro, a.dcjenis, a.kdcob,  
        b.insurance_contract_id_2,  
        a.dcdateloss, a.dccclaim,  
        a.dcsalvage, a.dcbefanclaim, a.kode_lokasi_afiliasi, a.sub_cob_id,  
a.idinsured, a.dctrxcode,
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
a.jenis_dctipedok, a.jenis_ddckot, a.coa_prefix_currency, a.subcob,
a.kob_cd, a.dcnoold, a.dctanggal
    from generate_endorsement3 a
    left join (
        select      distinct      dcno,      kdcob,      insurance_contract_id,
insurance_contract_id_2 from generate_endorsement3
            where insurance_contract_id is not null and insurance_contract_id != '_'
    ) b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno and a.kdcob = b.kdcob
    where a.insurance_contract_id is null or a.insurance_contract_id = '-'
),
generate_endorsement5 as (
    select * from generate_endorsement4 where insurance_contract_id is not null
and insurance_contract_id != '-'
    union all
    select
        a.reporting_dt, a.entity_id,
        b.insurance_contract_id,
        a.endorsement_type_cd, a.issue_dt, a.product_line_id, a.kob_id,
        a.product_id, a.source_of_business, a.ceding_id, a.cohort_id,
a.ceded_flg, a.reins_prop_cover_flg, a.reins_treaty_flg,
        a.endorsement_dt, a.begin_cov_dt, a.end_cov_dt, a.currency_cd,
a.prem_amt, a.base_alloc, a.comm_broker_amt, a.commission_amt,
        a.discount_amt, a.admin_fee_amt, a.pd, a.lgd,
a.transition_approach_cd, a.transition_cumulative_oci_amt,
a.year_defisit_clause,
        a.management_pct, a.profit_comm_pct, a.re_profit_comm_amt,
a.re_profit_comm_dt, a.source_data, a.category_data, a.dcno,
        a.dcinduk, a.dcstatus, a.dcnofile, a.dcnoretro, a.dcjenis, a.kdcob,
b.insurance_contract_id_2,
        a.dcdateloss, a.dccclaim,
        a.dcsalvage, a.dcbebanclaim, a.kode_lokasi_afiliasi, a.sub_cob_id,
a.idinsured, a.dctrxcode,
        a.jenis_dctipedok, a.jenis_ddckot, a.coa_prefix_currency, a.subcob,
a.kob_cd, a.dcnoold, a.dctanggal
    from generate_endorsement4 a
    left join (
        select      distinct      dcno,      kdcob,      insurance_contract_id,
insurance_contract_id_2 from generate_endorsement4
            where insurance_contract_id is not null and insurance_contract_id != '_'
    ) b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno and a.kdcob = b.kdcob
    where a.insurance_contract_id is null or a.insurance_contract_id = '-'
),
generate_endorsement6 as (
    select * from generate_endorsement5 where insurance_contract_id is not null
and insurance_contract_id != '-'
    union all
    select
        a.reporting_dt, a.entity_id,
        b.insurance_contract_id,
        a.endorsement_type_cd, a.issue_dt, a.product_line_id, a.kob_id,
        a.product_id, a.source_of_business, a.ceding_id, a.cohort_id,
a.ceded_flg, a.reins_prop_cover_flg, a.reins_treaty_flg,
        a.endorsement_dt, a.begin_cov_dt, a.end_cov_dt, a.currency_cd,
a.prem_amt, a.base_alloc, a.comm_broker_amt, a.commission_amt,
        a.discount_amt, a.admin_fee_amt, a.pd, a.lgd,
a.transition_approach_cd, a.transition_cumulative_oci_amt,
a.year_defisit_clause,
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
a.management_pct,      a.profit_comm_pct,      a.re_profit_comm_amt,
a.re_profit_comm_dt,  a.source_data,  a.category_data, a.dcno,
a.dcinduk, a.dcstatus, a.dcnofile, a.dcnoretro, a.dcjnis, a.kdcob,
b.insurance_contract_id_2,
a.dcdateloss, a.dccclaim,
a.dcsalvage, a.dcbefanclaim, a.kode_lokasi_afiliasi, a.sub_cob_id,
a.idinsured, a.dctrxcode,
a.jenis_dctipedok, a.jenis_ddckot, a.coa_prefix_currency, a.subcob,
a.kob_cd, a.dcnoold, a.dctanggal
from generate_endorsement5 a
left join (
    select      distinct      dcno,      kdcob,      insurance_contract_id,
insurance_contract_id_2 from generate_endorsement5
    where insurance_contract_id is not null and insurance_contract_id != '_'
)
b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno and a.kdcob = b.kdcob
where a.insurance_contract_id is null or a.insurance_contract_id = '-'
),
generate_endorsement7 as (
    select * from generate_endorsement6 where insurance_contract_id is not null
and insurance_contract_id != '-'
    union all
    select
        a.reporting_dt, a.entity_id,
        b.insurance_contract_id,
        a.endorsement_type_cd, a.issue_dt, a.product_line_id, a.kob_id,
        a.product_id, a.source_of_business, a.ceding_id, a.cohort_id,
        a.ceded_flg, a.reins_prop_cover_flg, a.reins_treaty_flg,
        a.endorsement_dt, a.begin_cov_dt, a.end_cov_dt, a.currency_cd,
        a.prem_amt, a.base_alloc, a.comm_broker_amt, a.commission_amt,
        a.discount_amt, a.admin_fee_amt, a.pd, a.lgd,
        a.transition_approach_cd, a.transition_cumulative_oci_amt,
        a.year_defisit_clause,
        a.management_pct, a.profit_comm_pct, a.re_profit_comm_amt,
a.re_profit_comm_dt, a.source_data, a.category_data, a.dcno,
a.dcinduk, a.dcstatus, a.dcnofile, a.dcnoretro, a.dcjnis, a.kdcob,
b.insurance_contract_id_2,
a.dcdateloss, a.dccclaim,
a.dcsalvage, a.dcbefanclaim, a.kode_lokasi_afiliasi, a.sub_cob_id,
a.idinsured, a.dctrxcode,
a.jenis_dctipedok, a.jenis_ddckot, a.coa_prefix_currency, a.subcob,
a.kob_cd, a.dcnoold, a.dctanggal
from generate_endorsement6 a
left join (
    select      distinct      dcno,      kdcob,      insurance_contract_id,
insurance_contract_id_2 from generate_endorsement6
    where insurance_contract_id is not null and insurance_contract_id != '_'
)
b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno and a.kdcob = b.kdcob
where a.insurance_contract_id is null or a.insurance_contract_id = '-'
),
latest_icid as (
    select
        coalesce(max(left(replace(insurance_contract_id,'-','0')), 5)::int),0
latest_icid
    from report.master_icid a
    where reporting_dt = '${reporting_dt}'
),
generate_icid_endorsement as (
    select
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
a.reporting_dt, a.entity_id,
coalesce(
    uu.insurance_contract_id,
    concat(
        to_char((row_number () over(order by a.begin_cov_dt asc)
+ (select max(latest_icid) from latest_icid)), 'fm00000'), -- Running Number
        to_char(date_part('MONTH',
 '${reporting_dt}')),'fm00'), -- Month
        a.kode_lokasi_afiliasi, -- Lokasi & Afiliasi
        a.kdcob, -- COB
        a.sub_cob_id, -- SUB COB

        to_char(date_part('YEAR',coalesce(a.begin_cov_dt,'1960-01-01'))),
        'fm0000'), -- Cohort
        a.source_of_business, -- Source of Business
        a.idinsured, -- Insured
        to_char(a.ceding_id::int,'fm000'), -- Ceding
        case
            when a.ceded_flg = 'N' then 'S'
            when a.ceded_flg = 'Y' then 'R'
            else '-'
        end, -- Insurance type
        a.kob_cd, -- KOB
        a.coa_prefix_currency -- Currency
    )
) as insurance_contract_id,
null as endorsement_type_cd, a.issue_dt, a.product_line_id, a.kob_id,
a.product_id, a.source_of_business, a.ceding_id, a.cohort_id,
a.ceded_flg, a.reins_prop_cover_flg, a.reins_treaty_flg,
null::date as endorsement_dt, a.begin_cov_dt, a.end_cov_dt,
a.currency_cd, a.prem_amt, a.base_alloc, a.comm_broker_amt, a.commission_amt,
a.discount_amt, a.admin_fee_amt, a.pd, a.lgd,
a.transition_approach_cd, a.transition_cumulative_oci_amt,
a.year_defisit_clause,
a.management_pct, a.profit_comm_pct, a.re_profit_comm_amt,
a.re_profit_comm_dt, a.source_data, a.category_data, a.dcno,
a.dcinduk, a.dccstatus, a.dcnofile, a.dcnoretro, a.dccjenis, a.kdcob,
coalesce(
    uu.insurance_contract_id_2,
    concat_ws(
        '|',
        to_char((row_number () over(order by a.begin_cov_dt asc)
+ (select max(latest_icid) from latest_icid)), 'fm00000'), -- Running Number
        to_char(date_part('MONTH',
 '${reporting_dt}')),'fm00'), -- Month
        a.kode_lokasi_afiliasi, -- Lokasi & Afiliasi
        a.kdcob, -- COB
        a.sub_cob_id, -- SUB COB

        to_char(date_part('YEAR',coalesce(a.begin_cov_dt,'1960-01-01'))),
        'fm0000'), -- Cohort
        a.source_of_business, -- Source of Business
        a.idinsured, -- Insured
        to_char(a.ceding_id::int,'fm000'), -- Ceding
        case
            when a.ceded_flg = 'N' then 'S'
            when a.ceded_flg = 'Y' then 'R'
            else '-'
        end, -- Insurance type
        a.kob_cd, -- KOB
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
a.coa_prefix_currency, -- Currency
    'NEWICID'
)
) as insurance_contract_id_2,
    a.dcdateloss,a.dcclaim,    a.dcsalvage,    a.dcbebanclaim,    a.subcob,
a.dcnold, a.dctanggal
from generate_endorsement7 a
left join report.master_icid uu on a.derno = uu.derno and a.kdcob = uu.kdcob
where a.insurance_contract_id is null or a.insurance_contract_id = '-' or
a.insurance_contract_id = ''
),
generate_endorsement8 as (
    select
        a.reporting_dt, a.entity_id,
        a.insurance_contract_id,
        a.endorsement_type_cd, a.issue_dt, a.product_line_id, a.kob_id,
        a.product_id,    a.source_of_business,    a.ceding_id,    a.cohort_id,
a.ceded_flg, a.reins_prop_cover_flg, a.reins_treaty_flg,
        a.endorsement_dt,    a.begin_cov_dt,    a.end_cov_dt,    a.currency_cd,
a.prem_amt, a.base_alloc, a.comm_broker_amt, a.commission_amt,
        a.discount_amt,           a.admin_fee_amt,           a.pd,           a.lgd,
a.transition_approach_cd,                      a.transition_cumulative_oci_amt,
a.year_defisit_clause,
        a.management_pct,      a.profit_comm_pct,      a.re_profit_comm_amt,
a.re_profit_comm_dt, a.source_data, a.category_data, a.derno,
        a.dcindeuk, a.dccstatus, a.dcnofile, a.dcnoretro, a.dcjenis, a.kdcob,
        a.insurance_contract_id_2,
        a.dcdateloss, a.dcclaim, a.dcsalvage, a.dcbebanclaim, a.subcob,
a.dcnold, a.dctanggal
from generate_endorsement7 a
where insurance_contract_id is not null and insurance_contract_id != '-' and
a.insurance_contract_id != ''
union all
    select
        a.reporting_dt, a.entity_id,
        a.insurance_contract_id,
        a.endorsement_type_cd, a.issue_dt, a.product_line_id, a.kob_id,
        a.product_id,    a.source_of_business,    a.ceding_id,    a.cohort_id,
a.ceded_flg, a.reins_prop_cover_flg, a.reins_treaty_flg,
        a.endorsement_dt,    a.begin_cov_dt,    a.end_cov_dt,    a.currency_cd,
a.prem_amt, a.base_alloc, a.comm_broker_amt, a.commission_amt,
        a.discount_amt,           a.admin_fee_amt,           a.pd,           a.lgd,
a.transition_approach_cd,                      a.transition_cumulative_oci_amt,
a.year_defisit_clause,
        a.management_pct,      a.profit_comm_pct,      a.re_profit_comm_amt,
a.re_profit_comm_dt, a.source_data, a.category_data, a.derno,
        a.dcindeuk, a.dccstatus, a.dcnofile, a.dcnoretro, a.dcjenis, a.kdcob,
        a.insurance_contract_id_2,
        a.dcdateloss, a.dcclaim, a.dcsalvage, a.dcbebanclaim, a.subcob,
a.dcnold, a.dctanggal
from generate_icid_endorsement a
),
dcno_kob2 as (
    select distinct
        aa.derno,
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.derno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end
)
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
end as kdcob,
case
    when aa.dctrxcode in ('11','51') then '5'
    when aa.dctrxcode = '22' then
        case
            when ii.jenis = 'SOA Life' then '4'
            else
                case
                    when nn.jenis = 'SOA Treaty QS'
                    when nn.jenis = 'SOA Treaty SPL'
                    when nn.jenis = 'SOA Treaty FACOB'
                    else '4'
                end
            end
        else '4'
    end kob_cd,
    case
        when aa.dctrxcode in ('11','51') then 'FC'
        when aa.dctrxcode = '22' then
            case
                when ii.jenis = 'SOA Life' then 'TN'
                else
                    case
                        when nn.jenis = 'SOA Treaty QS'
                        when nn.jenis = 'SOA Treaty SPL'
                        when nn.jenis = 'SOA Treaty FACOB'
                        else 'TN'
                    end
                end
            end
        else 'TN'
    end
    end as kob_id
from all_initial2 aa
left join staging.tbl_detaildncn gg on aa.dcno = gg.dcno
left join staging.ref_tipedok ii on aa.dctipedok = ii.code
left join staging.ref_tipedok nn on gg.ddckot = nn.code
),
generate_icid_initial as (
select
    '${reporting_dt}':date as reporting_dt,
    'TRE' as entity_id,
    coalesce(
        uu.insurance_contract_id,
        concat(
            to_char((row_number      ()      over(partition      by
date_part('YEAR',  '${reporting_dt}':date) order by aa.dcpriodestart asc)), 'fm00000'), -- Running Number
            to_char(date_part('MONTH',
 '${reporting_dt}':date), 'fm00'), -- Month
            ff.kode_lokasi_afiliasi, -- Lokasi & Afiliasi
            case
                when coalesce(gg.kdcob, aa.kdcob) = 'HE' then
'EN'
                when left(aa.dcno,2) in ('05','06') then
coalesce(gg.kdcob, aa.kdcob, 'LF')
            end
        )
    )
)
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        else coalesce(gg.kdcob, aa.kdcob)
    end, -- COB
    coalesce(kk.sub_cob_id,oo.sub_cob_id,'00'), -- SUB COB

        to_char(date_part('YEAR',coalesce(aa.dcperiodestart,'1960-01-01)::date)),
'fm0000'), -- Cohort
            ff.jenis, -- Source of Business
            coalesce(ee1.idinsured,ee2.idinsured), -- Insurred
            to_char(ff.kode_ifrs::int,'fm000'), -- Ceding
            case
                when bb.sesiretro = 'SESI' then 'S'
                when bb.sesiretro = 'RETRO' then 'R'
                else '-'
            end, -- Insurance type
            rr.kob_cd, -- KOB
            dd.coa_prefix -- Currency
        )
    ) as insurance_contract_id,
aa.endorsement_type as endorsement_type_cd,
coalesce(aa.dcperiodestart,'1960-01-01)::date) as issue_dt,
case
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
        else coalesce(gg.kdcob, aa.kdcob)
end as product_line_id,
rr.kob_id,
concat(
    coalesce(kk.sub_cob_id,oo.sub_cob_id,'00'),
    '_',
    rr.kob_id,
    '_',
    case
        when
            (
                (12 - EXTRACT(MONTH FROM dcperiodestart))
+
                (EXTRACT(MONTH FROM dcperiodeend)) +
                ((EXTRACT(YEAR FROM dcperiodeend) -
EXTRACT(YEAR FROM dcperiodestart) - 1) * 12) +
                (case when EXTRACT(DAY FROM dcperiodeend)
- EXTRACT(DAY FROM dcperiodeend) > 0 then 1 else 0 end)
            ) = 0 then 1
        else
            (
                (12 - EXTRACT(MONTH FROM dcperiodestart))
+
                (EXTRACT(MONTH FROM dcperiodeend)) +
                ((EXTRACT(YEAR FROM dcperiodeend) -
EXTRACT(YEAR FROM dcperiodestart) - 1) * 12) +
                (case when EXTRACT(DAY FROM dcperiodeend)
- EXTRACT(DAY FROM dcperiodeend) > 0 then 1 else 0 end)
            )
        end, 'mth',
        case
            when rr.kob_id in ('TQ','TS') then
concat('_',upper(qq.periode_pelaporan))
            when rr.kob_id in ('FC','TN') then
concat('_INSTALL',ss.n_installment,'_',tt.pattern)
            else ''
        end
    )
)
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        end
    ) as product_id,
    coalesce(ff.jenis,'') as source_of_business,
    to_char(ff.kode_ifrs::int,'fm000') as ceding_id,
    to_char(date_part('YEAR',coalesce(aa.dcp period start,'1960-01-01)::date)), 'fm0000') as cohort_id,
    case
        when bb.sesiretro = 'SESI' then 'N'
        when bb.sesiretro = 'RETRO' then 'Y'
        else '-'
    end as ceded_flg,
    case
        when bb.sesiretro = 'SESI' then 'N'
        else
            case
                when left(aa.dcno,2) in ('07','08') then
                    case
                        when gg.ddckot in
                            ('15','16','17') then 'Y'
                            else 'N'
                        end
                    else 'Y'
                end
            end as reins_prop_cover_flg,
            case
                when bb.sesiretro = 'SESI' then 'N'
                else
                    case
                        when bb."name" like '%Treaty%' then 'Y'
                        else 'N'
                    end
                end as reins_treaty_flg,
                null::date as endorsement_dt,
                coalesce(aa.dcp period start,'1960-01-01)::date) as begin_cov_dt,
                coalesce(aa.dcp period end,'1960-01-01)::date) as end_cov_dt,
                aa.dccurr as currency_cd,
                (
                    case
                        when gg.dcno is not null then
                            sum(coalesce(gg.ddc premium,0))
                            else sum(coalesce(aa.dcgross premium,0))
                        end
                    *
                    case
                        when left(aa.dcno,2) in ('07','08','05','06') then 1
                        when left(aa.dcno,2) in ('01','03') then 1
                        when left(aa.dcno,2) in ('02','04') then -1
                        else 0
                    end
                ) as prem_amt,
                case
                    when bb.sesiretro = 'RETRO' then 0
                    when ll.dcno_sesi is not null then
                        (
                            case
                                when gg.dcno is not null then
                                    sum(coalesce(gg.ddc premium,0))
                                    else sum(coalesce(aa.dcgross premium,0))
                                end
                            *
                        
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



1.0

23 Februari 2024

```
        case
          when      left(aa.dcno,2)      in
('07','08','05','06') then 1
          when left(aa.dcno,2) in ('01','03') then
          when left(aa.dcno,2) in ('02','04') then
          else 0
        end
      )
+
sum(ll.prem_amt_retro)
else
  case
    when gg.dcno is not null then
      sum(coalesce(gg.ddcpremium,0))
    end
  *
  case
    when left(aa.dcno,2) in ('07','08','05','06')
then 1
    when left(aa.dcno,2) in ('01','03') then 1
    when left(aa.dcno,2) in ('02','04') then -1
    else 0
  end
end as base_alloc,
(
  case
    when gg.dcno is not null then
      sum(coalesce(gg.ddcbrokerage,0))
    else sum(coalesce(aa.dcbrokerage,0))
  end
  *
  case
    when left(aa.dcno,2) in ('07','08','05','06') then 1
    when left(aa.dcno,2) in ('01','03') then -1
    when left(aa.dcno,2) in ('02','04') then 1
    else 0
  end
) as comm_broker_amt,
(
  case
    when gg.dcno is not null then
      sum(coalesce(gg.ddccommission,0))
    else sum(coalesce(aa.dccommission,0))
  end
  *
  case
    when left(aa.dcno,2) in ('07','08','05','06') then 1
    when left(aa.dcno,2) in ('01','03') then -1
    when left(aa.dcno,2) in ('02','04') then 1
    else 0
  end
) as commission_amt,
0 as discount_amt,
0 as admin_fee_amt,
case
  when bb.sesiretro = 'SESI' then 0
  else 0.0002
end
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
end as pd,
case
    when bb.sesiretro = 'SESI' then 0
    else 0.45
end as lgd,
'FRA' as transition_approach_cd,
0 as transition_cumulative_oci_amt,
max(qq.ydc) as year_defisit_clause,
max(qq.management_expenses) as management_pct,
max(qq.profit_commission_calc) as profit_comm_pct,
0 as re_profit_comm_amt,
null::date re_profit_comm_dt,
aa.category as source_data,
'INITIAL' as category_data,
trim(aa.dcno) dcno,
trim(aa.dcinduk) dcinduk,
trim(aa.dcstatus) dcstatus,
aa.dcnofile,
aa.dcnoretro,
aa.dcjenis,
case
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
        else coalesce(gg.kdcob, aa.kdcob)
end as kdcob,
coalesce(
    uu.insurance_contract_id_2,
    concat_ws(
        '|',
        to_char((row_number      ()      over(partition      by
date_part('YEAR', '${reporting_dt}':date) order by aa.dcp period start asc)),
'fm00000'), -- Running Number
        to_char(date_part('MONTH',
`${reporting_dt}`:date), 'fm00'), -- Month
        ff.kode_lokasi_afiliasi, -- Lokasi & Afiliasi
    case
        when coalesce(gg.kdcob, aa.kdcob) = 'HE' then
'EN'
        when left(aa.dcno,2) in ('05','06') then
coalesce(gg.kdcob, aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end, -- COB
        coalesce(kk.sub_cob_id,oo.sub_cob_id,'00'), -- SUB COB
        to_char(date_part('YEAR',coalesce(aa.dcp period start,'1960-01-01':date)),
'fm0000'), -- Cohort
        ff.jenis, -- Source of Business
        coalesce(ee1.idinsured,ee2.idinsured), -- Insured
        to_char(ff.kode_ifrs::int,'fm000'), -- Ceding
    case
        when bb.sesiretro = 'SESI' then 'S'
        when bb.sesiretro = 'RETRO' then 'R'
        else '-'
    end, -- Insurance type
    rr.kob_cd, -- KOB
    dd.coa_prefix -- Currency
)
) as insurance_contract_id_2,
aa.dcdateloss,
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
(  
    case  
        when gg.dcno is not null then  
            sum(coalesce(gg.ddcclaim,0))  
                else sum(coalesce(aa.dcclaim,0))  
            end  
        *  
    case  
        when left(aa.dcno,2) in ('07','08','05','06') then 1  
        when left(aa.dcno,2) in ('01','03') then -1  
        when left(aa.dcno,2) in ('02','04') then 1  
        else 0  
    end  
) as dcclaim,  
(  
    case  
        when gg.dcno is not null then  
            sum(coalesce(gg.ddsalvage,0))  
                else sum(coalesce(aa.dcsalvage,0))  
            end  
        *  
    case  
        when left(aa.dcno,2) in ('07','08','05','06') then 1  
        when left(aa.dcno,2) in ('01','03') then 1  
        when left(aa.dcno,2) in ('02','04') then -1  
        else 0  
    end  
) as dcsalvage,  
(  
    case  
        when gg.dcno is not null then  
            sum(coalesce(gg.ddcbebanclaim,0))  
                else sum(coalesce(aa.dcbebanclaim,0))  
            end  
        *  
    case  
        when left(aa.dcno,2) in ('07','08','05','06') then 1  
        when left(aa.dcno,2) in ('01','03') then -1  
        when left(aa.dcno,2) in ('02','04') then 1  
        else 0  
    end  
) as dcbebanclaim,  
aa.subcob,  
aa.dcnoold,  
aa.dctanggal  
from all_initial2 aa  
left join staging.ref_trxcode bb ON aa.dctrxcode = bb.code  
left join report.ref_currency dd ON aa.dccurr = dd.code  
left join (select * from staging.ref_insured where left(dcinsured,1) != '9') ee1 on aa.dcinsured = ee1.dcinsured  
left join (select * from staging.ref_insured where left(dcinsured,1) = '9') ee2 on aa.dcinsuredname = ee2.dcinsuredname  
left join staging.ref_ceding ff on aa.cedkd::int = ff.cedkd::int  
left join staging.tbl_detaildncn gg on aa.dcno = gg.dcno  
left join staging.ref_sub_cob kk on aa.subcob = kk.sub_cob_name and  
case  
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'  
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,  
aa.kdcob, 'LF')  
    else coalesce(gg.kdcob, aa.kdcob)
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
        end = kk.kdcob
    left join link_sesi_retro ll on aa.dcno = ll.dcno_sesi and aa.dcnofile =
ll.dcnofile and aa.kdcob = ll.kdcob and aa.subcob = ll.subcob and aa.dccurr =
ll.dccurr
    left join (select kdcob, min(sub_cob_id) as sub_cob_id from
staging.ref_sub_cob group by kdcob) oo on
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end = oo.kdcob
    left join staging.tbl_jurnal pp on aa.dcno = pp.jnnonota
    left join (
        select
            (cob_item::json->0->'written_share'-'ydc')::text::float8 as
ydc,
            (cob_item::json->0->'written_share'-
>'management_expenses')::text::float8 management_expenses,
            (cob_item::json->0->'written_share'-
>'profit_commission_calc')::text::float8 profit_commission_calc,
            replace((bordereauxaccounts::json->'accounts')::text,'','','')
periode_pelaporan,
            treaty_number
        from staging.treaties
    ) qq on aa.dcnofile = qq.treaty_number
    left join dcno_kob2 rr on aa.dcno = rr.dcno and
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end = rr.kdcob
    left join (
        select dcno, count(*) n_installment from staging.tbl_installdncn ti
group by dcno
    ) ss on aa.dcno = ss.dcno
    left join staging.ref_pattern_product_id tt on
        ss.n_installment = tt.installment and
        coalesce(aa.dcwp, DATE_PART('day', dcdue date)::timestamp) -
dcperiodestart::timestamp,0) >= tt.wpc_min and
        coalesce(aa.dcwp, DATE_PART('day', dcdue date)::timestamp) -
dcperiodestart::timestamp,0) <= tt.wpc_max
    left join report.master_icid uu on aa.dcno = uu.dcno and
        case
            when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
            when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
            else coalesce(gg.kdcob, aa.kdcob)
        end = uu.kdcob
    where 1=1
        and pp.jnnstposting = '1'
        and aa.dcstsjurnal = '4'
    group by
        aa.dcno,aa.dcperiodestart,aa.dcendorsement_type,aa.dctanggal,aa.dcperiodeend,
        ,aa.dccurr,aa.category,
        aa.dcinduk,aa.dcstatus,aa.dcnofile,aa.dcnostretro,aa.dcjenis,aa.dcdateloss,a
        a.subcob,aa.dcnoold,
```

```

bb.sesiretro,bb.name,
dd.coa_prefix,
ee1.idinsured,
ee2.idinsured,
ff.kode_lokasi_afiliasi,ff.jenis,ff.kode_ifrs,
gg.ddckot,gg.dcno,
kk.sub_cob_id,
ll.dcno_sesi,
oo.sub_cob_id,
qq.periode_pelaporan,
rr.kob_cd,rr.kob_id,
ss.n_installment,
tt.pattern,
uu.insurance_contract_id,uu.insurance_contract_id_2,
case
    when coalesce(gg.kdcob, aa.kdcob) = 'HE' then 'EN'
    when left(aa.dcno,2) in ('05','06') then coalesce(gg.kdcob,
aa.kdcob, 'LF')
    else coalesce(gg.kdcob, aa.kdcob)
end
),
union_all_data as (
    select * from generate_icid_initial
    union all
    select * from generate_endorsement8
)
select * from union_all_data a

```

- PutDatabaseRecord



Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel insurance_contract di database PostgreSQL DWH.

- d. Filter COB and SUBCOB

Pada proses ini akan memfilter data initial dan endorsement di tabel insurance_contract, hanya diambil COB *MH, FR, MV* dan COB *EN* hanya ambil SUBCOB *CONTRACTOR ALL RISK* dan *ERETION ALL RISK*.

Processor yang digunakan untuk step adalah sebagai berikut:

- ExecuteSQL



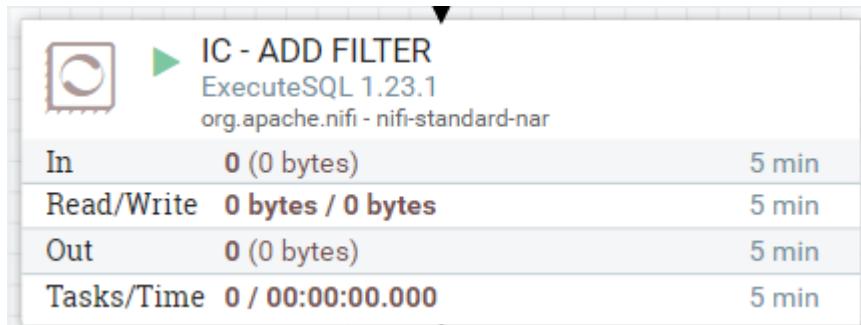
TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024



Processor ini digunakan untuk mengeksekusi perintah SQL untuk menambahkan filter COB *MH*, *FR*, *MV* dan COB *EN* hanya ambil SUBCOB *CONTRACTOR ALL RISK* dan *ERETCTION ALL RISK* dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

Berikut query yang digunakan untuk menambahkan filter tersebut:

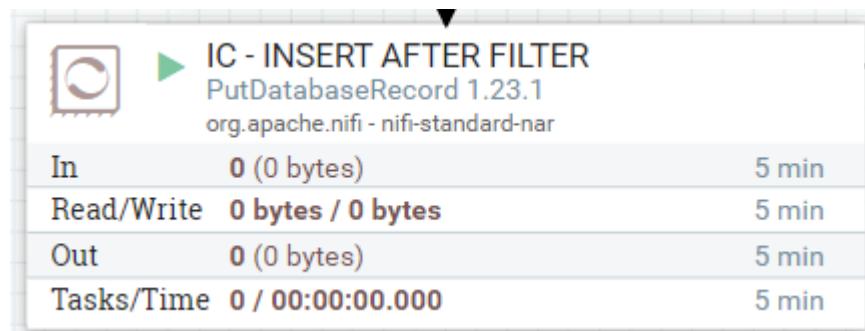
```
with insurance_contract_update as (
    select
        '${reporting_dt}':date as reporting_dt,
        entity_id,
        a.insurance_contract_id,
        case
            when (a.category_data = 'ENDORSEMENT' and
a.insurance_contract_id_2 not like '%NEWICID%') or endorsement_type_cd = '99'
            then
                case
                    when c.category_data = 'INITIAL' and a.prem_amt +
c.prem_amt = 0 then '03'
                    when c.category_data = 'ENDORSEMENT' and
a.dcnofile = c.dcnofile and a.prem_amt + c.prem_amt = 0 then '06'
                    when a.end_cov_dt = c.end_cov_dt or a.end_cov_dt =
c.end_cov_dt then '01'
                    when a.end_cov_dt != c.end_cov_dt then '02'
                    else '01'
                end
            else null
        end as endorsement_type_cd,
        coalesce(d.issue_dt,a.issue_dt) as issue_dt,
        product_line_id, kob_id, product_id,
        source_of_business, ceding_id,
        coalesce(d.cohort_id, a.cohort_id) as cohort_id,
        ceded_flg, reins_prop_cover_flg, reins_treaty_flg, endorsement_dt,
        coalesce(d.begin_cov_dt,a.begin_cov_dt) as begin_cov_dt,
        a.end_cov_dt,
        currency_cd, a.prem_amt, base_alloc, comm_broker_amt, commission_amt,
        discount_amt, admin_fee_amt, pd, lgd, transition_approach_cd,
        transition_cumulative_oci_amt, year_defisit_clause, management_pct,
        profit_comm_pct, re_profit_comm_amt, re_profit_comm_dt,
        source_data, a.category_data, a.dcnno, dcinduk, dcstatus, a.dcnofile,
        dcnotaretro, dcjenis, a.kdcob, a.insurance_contract_id_2,
        dcdateloss, dcclaim, dcsalvage, dcbebanclaim, subcob, dcnoold, dctanggal,
        reporting_dt as reporting_dt2
        from report.${insurance_contract_step1} a
        left join (
            select distinct category_data, prem_amt, dcno, kdcob, dcnofile,
            end_cov_dt
            from report.${insurance_contract_step1}
        ) c on trim(SPLIT_PART(a.dcinduk, '|', 1)) = c.dcnno and a.kdcob = c.kdcob
```

```

    left join (
        select distinct insurance_contract_id, begin_cov_dt, issue_dt,
cohort_id
        from report.${insurance_contract_step1} where category_data = 'INITIAL'
    ) d on a.insurance_contract_id = d.insurance_contract_id
    where 1=1
),
insurance_contract_update2 as (
    select * from insurance_contract_update where endorsement_dt is null
    union all
    select * from insurance_contract_update where endorsement_dt is not null and
endorsement_dt <= reporting_dt
),
icid_active as (
    select distinct insurance_contract_id from insurance_contract_update2 ich
    where (
        (begin_cov_dt <= '${year_reporting_dt}-01-01' and end_cov_dt >=
 '${year_reporting_dt}-01-01')
        or
        (begin_cov_dt > '${year_reporting_dt}-01-01' and begin_cov_dt
<= '${reporting_dt}' and end_cov_dt >= '${reporting_dt}')
    )
        and cohort_id::int >= 2020
),
insurance_contract as (
    select
        a.*
    from insurance_contract_update a
    join icid_active b on a.insurance_contract_id = b.insurance_contract_id
    where 1=1
        and a.issue_dt <= '${reporting_dt}'
        and (
            product_line_id in ('MH','FR','MV')
            or (product_line_id = 'EN' and subcob in ('CONTRACTOR ALL
RISK','ERECITION ALL RISK'))
        )
        and kob_id = 'FC'
        and dctanggal <= '${reporting_dt}'
)
select * from insurance_contract

```

- PutDatabaseRecord



Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel insurance_contract di database PostgreSQL DWH.

- Remove contract with ENDORSEMENT_DT > REPORTING_DT



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0

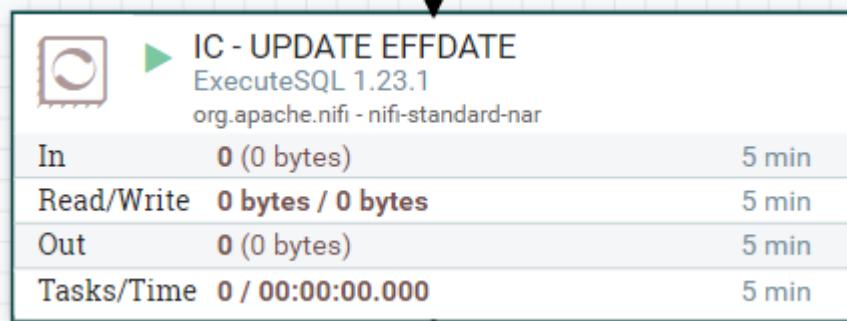


23 Februari 2024

Proses ini bertujuan untuk mengupdate endorsement_dt kontrak Endorsement dengan effective_date dari kontrak Initialnya, menghapus kontrak yang statusnya Cancel Endorsement (06) dan menghapus kontrak yang endorsement_dt nya lebih besar dari reporting_dt.

Processor yang digunakan untuk step adalah sebagai berikut:

- ExecuteSQL



Processor ini digunakan untuk mengeksekusi perintah SQL untuk mengupdate endorsement_dt kontrak Endorsement dengan effective_date dari kontrak Initialnya, menghapus kontrak yang statusnya Cancel Endorsement (06) dan menghapus kontrak yang endorsement_dt nya lebih besar dari reporting_dt dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

Berikut query yang digunakan untuk menambahkan filter tersebut:

```
with effdate as (
    select a.dcno, max(b.transaction_id) transaction_id from staging.tbl_dncn a
    join staging.note_numbers b on a.dcno = b."number" and a.dccurr = b.currency
    where b.transaction_id not like '%-old%'
    group by dcno
),
update_effdate as (
    select
        a.reporting_dt,           a.entity_id,           a.insurance_contract_id,
        a.endorsement_type_cd,   a.issue_dt,
        a.product_line_id,       a.kob_id,      a.product_id,   a.source_of_business,
        a.ceding_id,             a.cohort_id,
        a.ceded_flg,             a.reins_prop_cover_flg, a.reins_treaty_flg,
        case when c.effective_date is not null then
            case when c.effective_date < a.begin_cov_dt then
                a.endorsement_dt
            else c.effective_date
            end
        else a.endorsement_dt
        end as endorsement_dt,
        a.begin_cov_dt,
        a.end_cov_dt,           a.currency_cd,          a.prem_amt,      a.base_alloc,
        a.comm_broker_amt,       a.commission_amt,
        a.discount_amt,          a.admin_fee_amt,        a.pd,          a.lgd,
        a.transition_approach_cd, a.transition_cumulative_oci_amt,
        a.year_defisit_clause,   a.management_pct,      a.profit_comm_pct,
        a.re_profit_comm_amt,    a.re_profit_comm_dt,
        a.source_data,           a.category_data,     a.dcno,       a.dcinduk,     a.dcstatus,
        a.dcnofile,              a.dcnoretro,
        a.dcjenis,               a.kdcob,           a.insurance_contract_id_2,   a.dcdateloss,
        a.dclaim,                 a.dcsalvage,         a.dcbebanclaim,
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
a.subcob, a.dcnoold, a.dctanggal, a.dwh_created_date
from report.${insurance_contract_step2} a
left join effdate b on a.dcno = b.dcno
left join staging.transactions c on b.transaction_id = c.id
where endorsement_dt is not null
union all
select * from report.${insurance_contract_step2}
where endorsement_dt is null
),
dcno_remove as (
    select distinct concat(a.dcno,'#',a.kdcob) as dcno_kdcob from update_effdate
a
    where a.endorsement_type_cd = '06'
union all
    select distinct concat(b.dcno,'#',b.kdcob) as dcno_kdcob from update_effdate
a
    join update_effdate b on trim(SPLIT_PART(a.dcinduk, '|', 1)) = b.dcno and
a.kdcob = b.kdcob
        where a.endorsement_type_cd = '06'
),
remove_endorsement_cancellation as (
    select a.* from update_effdate a
    where concat(a.dcno,'#',a.kdcob) not in (select distinct dcno_kdcob from
dcno_remove)
),
remove_endorsement_dt_over_reporting_dt as (
    select * from remove_endorsement_cancellation
    where endorsement_dt <= reporting_dt and endorsement_dt is not null
    union all
    select * from remove_endorsement_cancellation
    where endorsement_dt is null
)
select * from remove_endorsement_dt_over_reporting_dt;
```

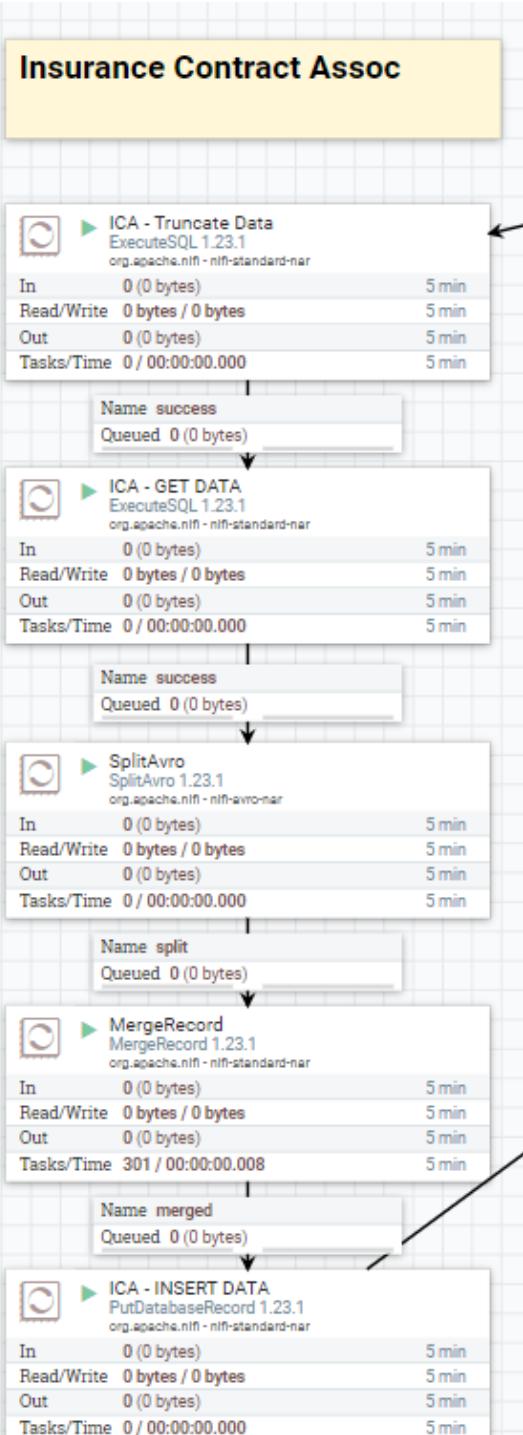
- PutDatabaseRecord

IC - INSERT AFTER UPDATE EFFDATE
PutDatabaseRecord 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel insurance_contract di database PostgreSQL DWH.

3. Proses Insurance Contract Assoc



Processor yang digunakan untuk proses insurance contrat assoc adalah sebagai berikut:

- ExecuteSQL



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengeksekusi perintah SQL untuk mendapatkan data assoc berdasarkan sumber data insurance_contract dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

Berikut query yang digunakan untuk menambahkan filter tersebut:

```
with ic_sesi as (
    select distinct reporting_dt, a.dcno, dcnofile, insurance_contract_id,
    dcnoretro, entity_id, kdcob, prem_amt, subcob, currency_cd
    from report.${insurance_contract} a
    where a.dcno like '%-K11%'
),
ic_retro as (
    select distinct reporting_dt, a.dcno, dcnofile, insurance_contract_id,
    dcnoretro, entity_id, kdcob, prem_amt, subcob, currency_cd
    from report.${insurance_contract} a
    where a.dcno like '%-K51%'
),
insurance_contract_assoc_dwh as (
    select distinct
        a.dcno as dcno_sesi,
        b.dcno as dcno_retro,
        a.reporting_dt,
        a.entity_id,
        a.insurance_contract_id,
        b.insurance_contract_id as rel_insurance_contract_id,
        1 as weight_pct,
        case
            when abs(b.prem_amt)/abs(a.prem_amt) > 1 then 1
            when abs(b.prem_amt)/abs(a.prem_amt) < 0 then 0
            else abs(b.prem_amt)/abs(a.prem_amt)
        end as reins_fixed_recover_pct
    from ic_sesi a
    join ic_retro b on a.dcnofile = b.dcnofile and a.kdcob = b.kdcob and a.subcob
    = b.subcob and a.currency_cd = b.currency_cd
    where a.dcnofile is not null and a.dcnofile != '' and b.dcnofile is not null
    and b.dcnofile != ''
),
insurance_contract_assoc_manual as (
    select distinct
        a.dcno_sesi,
        a.dcno as dcno_retro,
        b.reporting_dt,
        b.entity_id,
        b.insurance_contract_id,
        c.insurance_contract_id as rel_insurance_contract_id,
        1 as weight_pct,
        case
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

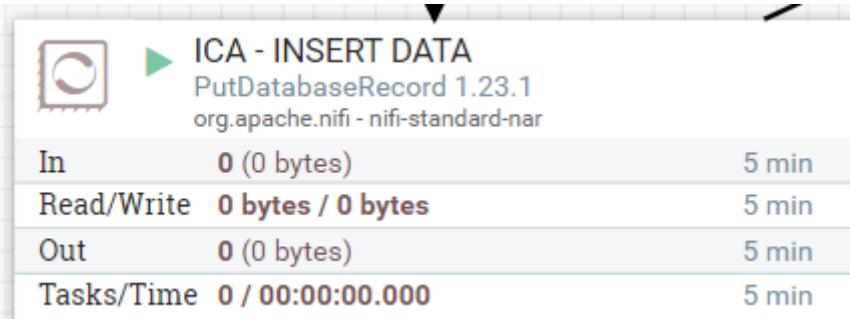


23 Februari 2024

```
        when abs(c.prem_amt)/abs(b.prem_amt) > 1 then 1
        when abs(c.prem_amt)/abs(b.prem_amt) < 0 then 0
        else abs(c.prem_amt)/abs(b.prem_amt)
    end as reins_fixed_recover_pct
from (
    select distinct dcno_sesi, dcno, dcnofile, kdcob, subcob, dcurr
    from report.dcno_retro_not_connected
    where dcno_sesi is not null or dcno_sesi != ''
) a
left join report.${insurance_contract} b on a.dcno_sesi = b.dcno
left join report.${insurance_contract} c on a.dcno = c.dcno
),
insurance_contract_assoc1 as (
    select distinct
        coalesce(a.dcno_sesi,b.dcno_sesi) as dcno,
        coalesce(a.reporting_dt,b.reporting_dt) as reporting_dt,
        coalesce(a.entity_id,b.entity_id) as entity_id,
        coalesce(a.insurance_contract_id,b.insurance_contract_id) as insurance_contract_id,
        coalesce(a.rel_insurance_contract_id,b.rel_insurance_contract_id) as rel_insurance_contract_id,
        coalesce(a.weight_pct,b.weight_pct) as weight_pct,
        coalesce(a.reins_fixed_recover_pct,b.reins_fixed_recover_pct) as reins_fixed_recover_pct
        from insurance_contract_assoc_dwh a
        full outer join insurance_contract_assoc_manual b on a.dcno_sesi = b.dcno_sesi
        and a.dcno_retro = b.dcno_retro
),
insurance_contract_dist as (
    select distinct insurance_contract_id, dcno, reporting_dt from report.${insurance_contract}
),
insurance_contract_assoc2 as (
    select
        '${reporting_dt}'::date as reporting_dt,
        'TRE' as entity_id,
        b.insurance_contract_id as insurance_contract_id,
        c.insurance_contract_id as rel_insurance_contract_id,
        1 as weight_pct,
        0 as reins_fixed_recover_pct
    from (
        select distinct nomor_nota_facultative, nomor_nota_retro
        from staging.ref_sesi_retro_edoc a
        where nomor_nota_facultative not in (select distinct dcno from insurance_contract_assoc1)
    ) a
    left join insurance_contract_dist b on trim(a.nomor_nota_facultative) = trim(b.dcno)
    left join insurance_contract_dist c on trim(a.nomor_nota_retro) = trim(c.dcno)
    where nomor_nota_facultative is not null and nomor_nota_facultative != ''
        and nomor_nota_facultative not like '%ERROR%'
        and b.dcno is not null and c.dcno is not null
),
insurance_contract_assoc as (
    select
        reporting_dt,
        entity_id,
        insurance_contract_id,
        rel_insurance_contract_id,
        weight_pct,
```

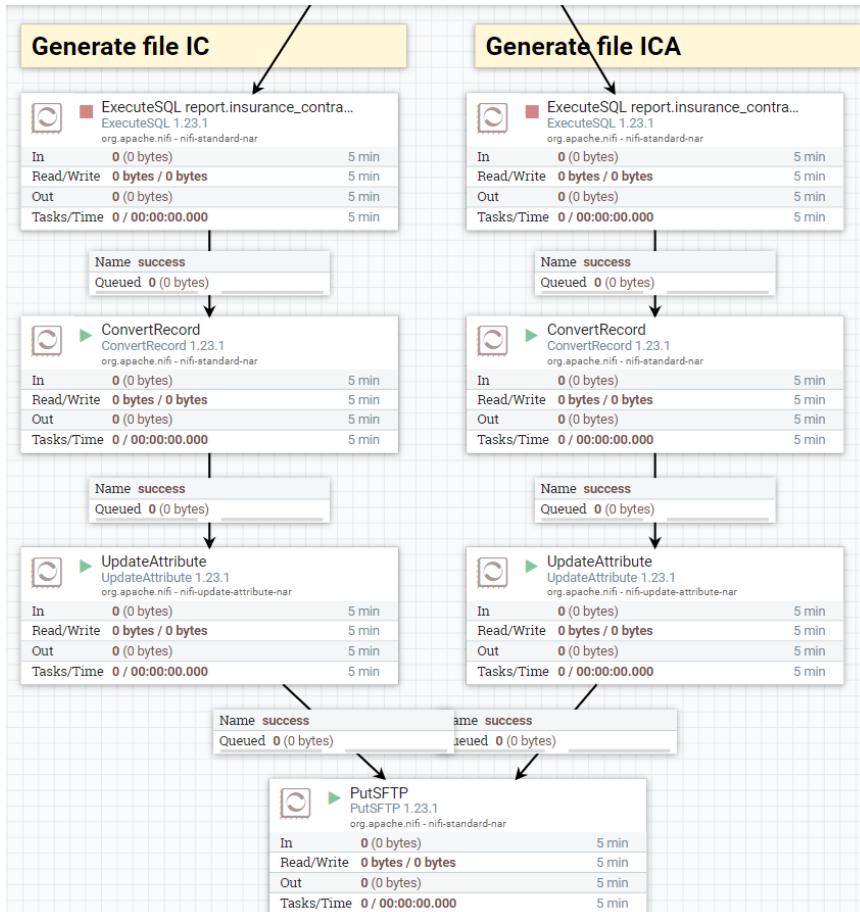
```
max(coalesce(reins_fixed_recover_pct,0)) as reins_fixed_recover_pct
from (
    select
reporting_dt,entity_id,insurance_contract_id,rel_insurance_contract_id,weight_pct,
reins_fixed_recover_pct
        from insurance_contract_assoc1
        union all
        select
reporting_dt,entity_id,insurance_contract_id,rel_insurance_contract_id,weight_pct,
reins_fixed_recover_pct
        from insurance_contract_assoc2
    ) x
    group      by      reporting_dt,      entity_id,      insurance_contract_id,
rel_insurance_contract_id, weight_pct
)
select * from insurance_contract_assoc;
```

- PutDatabaseRecord



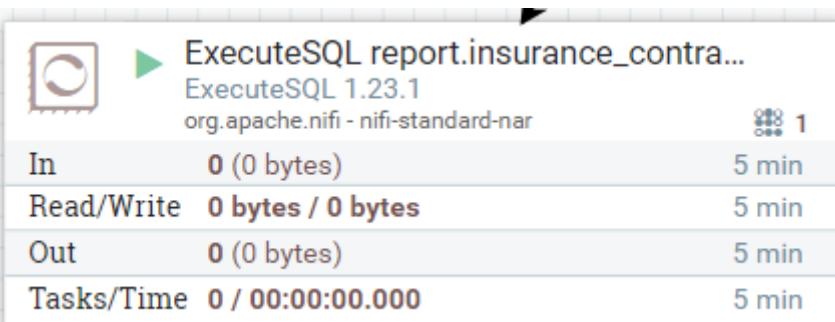
Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel `insurance_contract_assoc` di database PostgreSQL DWH.

4. Generate file insurance_contract.csv dan insurance_contract_assoc.csv



Processor yang digunakan untuk proses generate insurance_contract.csv dan insurance_contract_assoc.csv adalah sebagai berikut:

- ExecuteSQL



Processor ini digunakan untuk mengeksekusi perintah SQL untuk mendapatkan data insurance_contract dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

Berikut konfigurasi di dalam processor ExecuteSQL:



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

Processor Details | ExecuteSQL 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Database Connection Pooling Service	DBCPConnectionPool - tugure_dwh
SQL Pre-Query	No value set
SQL select query	select * from report.insurance_contract;
SQL Post-Query	No value set
Max Wait Time	0 seconds
Normalize Table/Column Names	false
Use Avro Logical Types	false
Compression Format	NONE
Default Decimal Precision	10
Default Decimal Scale	0
Max Rows Per Flow File	0
Output Batch Size	0

OK

- ConvertRecord

ConvertRecord
ConvertRecord 1.23.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk mengubah output flowfile dengan format tertentu menjadi format yang berbeda. Pada process group ini, ConvertRecord digunakan untuk mengubah flowfile dari Avro menjadi CSV.

Berikut konfigurasi di dalam processor ConvertRecord:



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

Processor Details | ConvertRecord 1.23.1

▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Record Reader	AvroReader
Record Writer	CSVRecordSetWriter
Include Zero Record FlowFiles	true

- UpdateAttribute

UpdateAttribute
UpdateAttribute 1.23.1
org.apache.nifi - nifi-update-attribute-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor ini digunakan untuk menambahkan atau mengubah attribute pada suatu flowfile, pada process group ini UpdateAttribute digunakan untuk menambahkan attribute filename pada flowfile NiFi.

Berikut konfigurasi di dalam processor UpdateAttribute:

Processor Details | UpdateAttribute 1.23.1

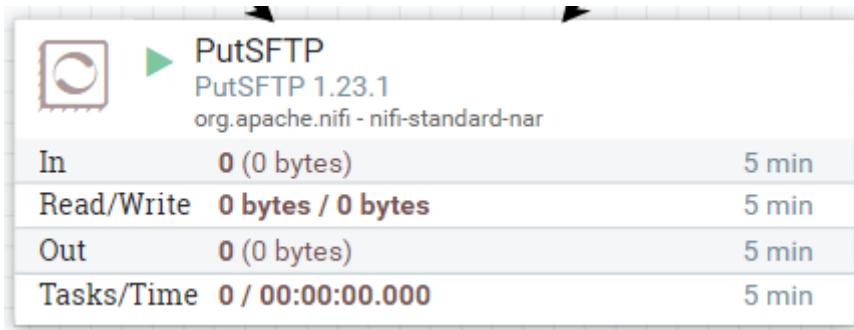
▶ Running STOP & CONFIGURE

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

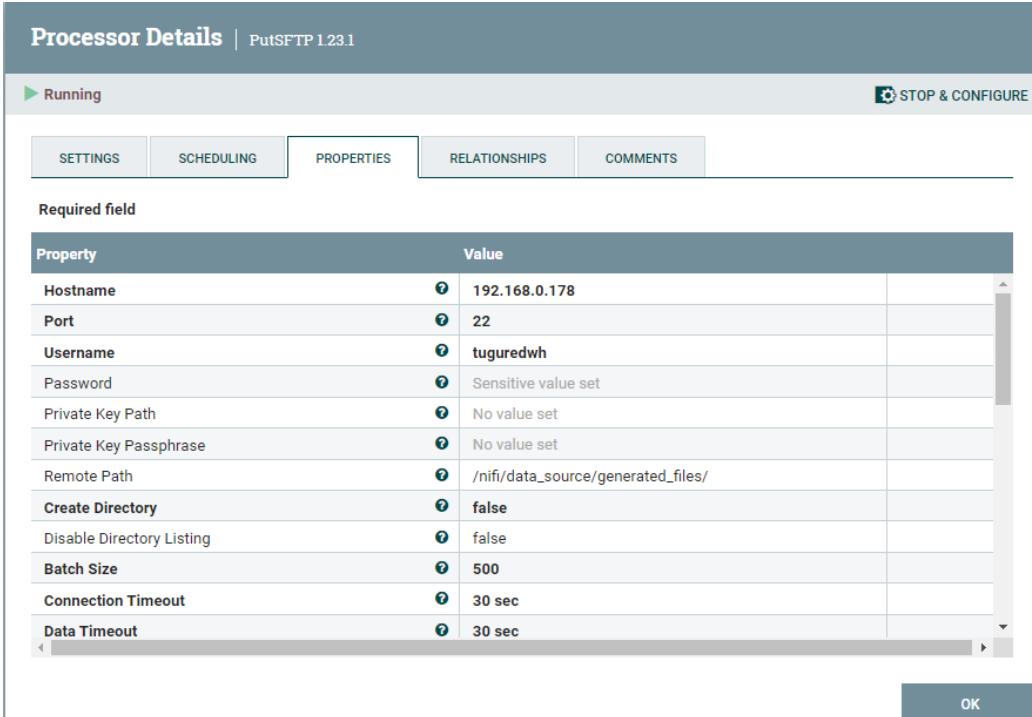
Property	Value
Delete Attributes Expression	No value set
Store State	Do not store state
Stateful Variables Initial Value	No value set
Cache Value Lookup Cache Size	100
filename	insurance_contract.csv

- PutSFTP



Processor ini digunakan untuk mengunggah (upload) file dari aliran data (flowfile) ke sebuah server SFTP (Secure File Transfer Protocol).

Berikut konfigurasi di dalam processor PutSFTP:



Property	Value
Hostname	192.168.0.178
Port	22
Username	tuguredwh
Password	Sensitive value set
Private Key Path	No value set
Private Key Passphrase	No value set
Remote Path	/nifi/data_source/generated_files/
Create Directory	false
Disable Directory Listing	false
Batch Size	500
Connection Timeout	30 sec
Data Timeout	30 sec

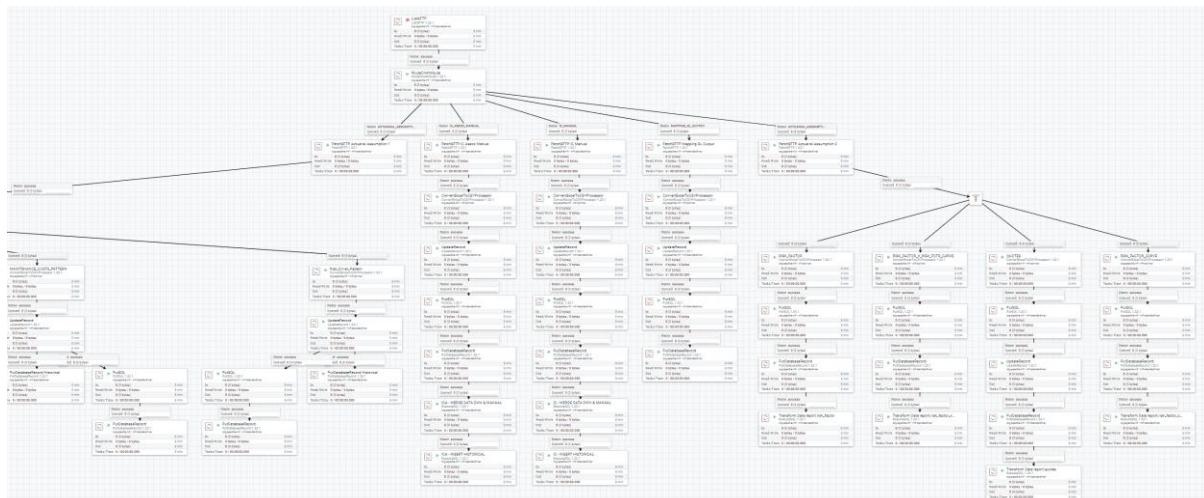
Pipeline 3

Pada Process Group Pipeline 3 berisi Processor untuk melakukan ingestion data dari file yang dikeluarkan oleh Tugure. Hasil ingestion dari file tersebut selanjutnya akan disimpan di database PostgreSQL dan berjalan setiap bulan mulai tanggal 22 sampai dengan akhir bulan jam 01.00 WIB.

Pipeline 3

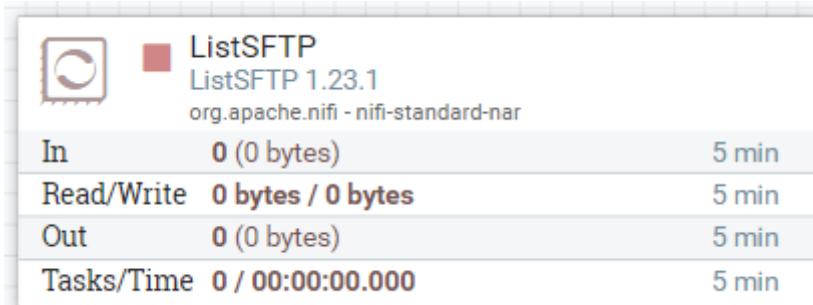
⌚ 0 ⌚ 0 ▶ 74 🔴 1 ⚠ 0 ✖ 0

Queued	0 (0 bytes)
In	0 (0 bytes) → 0
Read/Write	0 bytes / 0 bytes
Out	0 → 0 (0 bytes)

✓ 0 * 0 ⬆ 0 ❗ 0 ❓ 0


Di dalam Process Group Pipeline 3 terdapat beberapa proses yaitu:

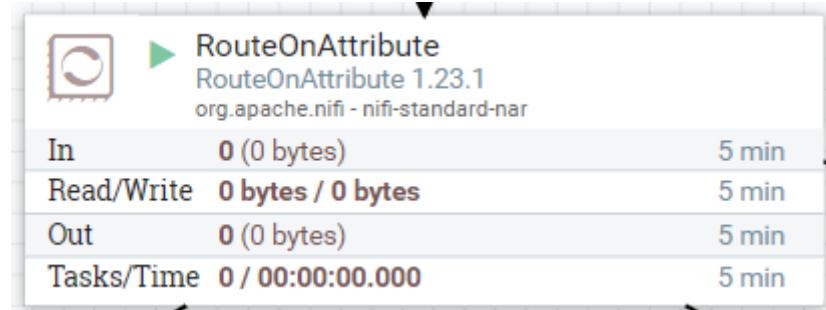
1. *Proses Ingestion File Actuarial Assumption 1, Actuarial Assumption 2, Insurance Contract Assoc Manual, Insurance Contract Manual, dan Mapping GL Output*
 - ListSFTP



Processor ini berfungsi untuk mendapatkan daftar file dari sebuah server SFTP (Secure File Transfer Protocol), processor ini digunakan untuk mendapatkan file Actuarial

Assumption 1, Actuarial Assumption 2, Insurance Contract Assoc Manual, Insurance Contract Manual, dan Mapping GL Output.

- RouteOnAttribute

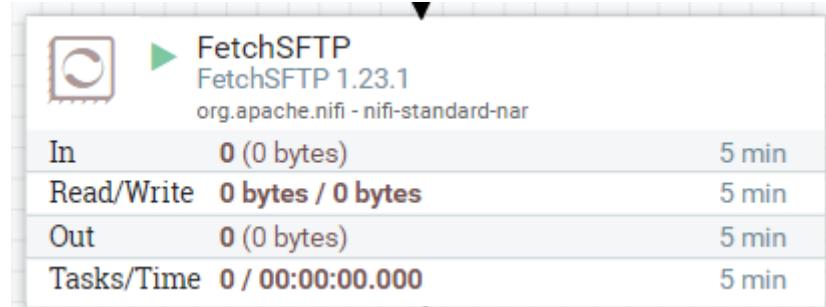


Processor ini digunakan untuk memecah aliran data (flowfile) ke beberapa processor tujuan berbeda berdasarkan nilai atribut yang ditentukan pada processor RouteOnAttribute.

Property	Value
Routing Strategy	Route to Property name
ACTUARIAL_ASSUMPTION_1	\$(filename:contains('Actuarial Assumption_1'))
ACTUARIAL_ASSUMPTION_2	\$(filename:contains('Actuarial Assumption_2'))
IC_ASSOC_MANUAL	\$(filename:contains('insurance_contract_assoc_manual'))
IC_MANUAL	\$(filename:contains('insurance_contract_manual'))
MAPPING_GL_OUTPUT	\$(filename:contains('Accounting Data'))

Disini processor membagi flowfile menjadi 5 yaitu ACTUARIAL_ASSUMPTION_1, ACTUARIAL_ASSUMPTION_2, IC_ASSOC_MANUAL, IC_MANUAL, dan MAPPING_GL_OUTPUT.

- FetchSFTP



Processor ini digunakan untuk mengunduh (fetch) file dari server SFTP (Secure File Transfer Protocol), mengambil datanya untuk kemudian dijadikan aliran data (flowfile).

- PutDatabaseRecord



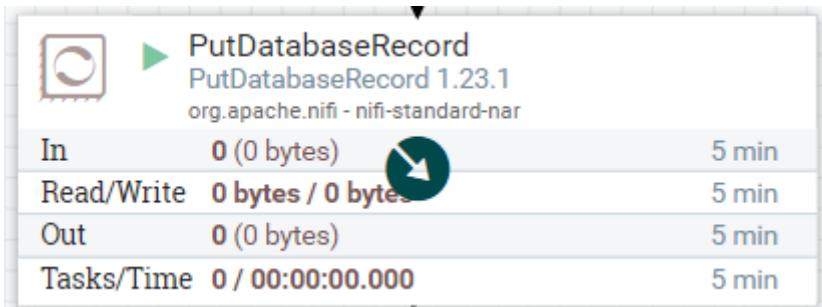
1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024



Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel di database PostgreSQL DWH.

- ConvertExcelToCSVProcessor

	ConvertExcelToCSVProcessor	ConvertExcelToCSVProcessor 1.23.1	org.apache.nifi - nifi-poi-nar
In	0 (0 bytes)	5 min	
Read/Write	0 bytes / 0 bytes	5 min	
Out	0 (0 bytes)	5 min	
Tasks/Time	0 / 00:00:00.000	5 min	

Processor ini digunakan untuk melakukan konversi terhadap aliran data (flowfile) yang berbentuk excel ke format csv agar lebih mudah diproses oleh processor Nifi.

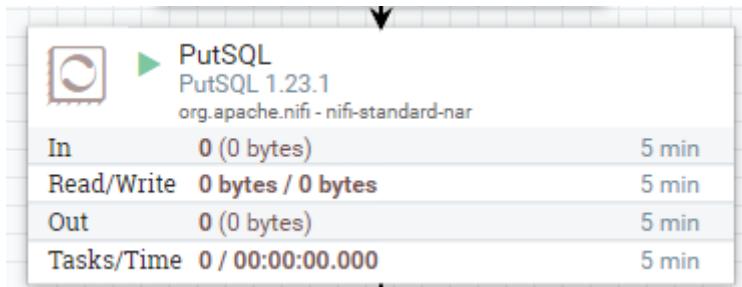
- UpdateRecord

	UpdateRecord	UpdateRecord 1.23.1	org.apache.nifi - nifi-standard-nar
In	0 (0 bytes)	5 min	
Read/Write	0 bytes / 0 bytes	5 min	
Out	0 (0 bytes)	5 min	
Tasks/Time	0 / 00:00:00.000	5 min	

Processor ini digunakan untuk memperbarui atau menambahkan record pada setiap FlowFile yang melewati prosesor tersebut dalam aliran data (flowfile). Didalam processor ini attribut yang diperbarui atau ditambahkan adalah sebagai berikut:

- insurance_contract_assoc_manual:
 - Mengubah format tanggal reporting_dt dari yyyyymmdd menjadi yyyy-mm-dd
- insurance_contract_manual:
 - Mengubah format tanggal reporting_dt, begin_cov_dt, end_cov_dt, issue_date dari yyyyymmdd menjadi yyyy-mm-dd
 - Menambahkan data category_data dengan value 'INITIAL'
 - Menambahkan data source_data dengan value 'Manual'
- Mapping_gl_output:
 - Mengubah format tanggal last_update dari yyyyymmdd menjadi yyyy-mm-dd
- Actuarial_assumption_1:
 - Mengubah format tanggal maturity_dt dan quote_dt dari yyyyymmdd menjadi yyyy-mm-dd
- Actuarial_assumption_2:
 - Mengubah format tanggal reporting_dt dari yyyyymmdd menjadi yyyy-mm-dd
 - Mengubah data karakter pada lfrc_claims_ratio yang sebelumnya menggunakan ',' (comma) menjadi ` (period).

- PutSQL



Processor ini digunakan untuk mengeksekusi perintah SQL ke database PostgreSQL DWH, perintah SQL nya untuk menghapus (truncate) data dari tabel di database PostgreSQL DWH.

2. Proses Transformasi Data

- ExecuteSQL

Processor ini digunakan untuk mengeksekusi perintah SQL ke database PostgreSQL DWH, perintah-perintah SQL yang dieksekusi adalah sebagai berikut:

- Menggabungkan Insurance Contract Assoc Manual dengan Insurance Contract Assoc:

```
insert into report.insurance_contract_assoc select * from
staging.insurance_contract_assoc_manual
```

- Menggabungkan Insurance Contract Manual dengan Insurance Contract:

```
insert into report.insurance_contract select reporting_dt, entity_id,
insurance_contract_id, endorsement_type_cd, issue_dt, product_line_id, kob_id,
product_id, source_of_business, ceding_id, cohort_id, ceded_flg,
reins_prop_cover_flg, reins_treaty_flg, endorsement_dt, begin_cov_dt,
end_cov_dt, currency_cd, prem_amt, base_alloc, comm_broker_amt,
commission_amt, discount_amt, admin_fee_amt, pd, lgd,
transition_approach_cd, transition_cumulative_oci_amt, year_defisit_clause,
management_pct, profit_comm_pct, re_profit_comm_amt, re_profit_comm_dt,
source_data, category_data, null as dcno, null as dcinduk, null as dcstatus, null as
dcnofile, null as dcnotaretro, null as dcjenis, null as kdcob, null as
insurance_contract_id_2, null::date as dcdateloss, null as dcclaim, null as
dcsalvage, null as dcbebanclaim, null as subcob, null as dcnoold, null::date as
dctanggal FROM staging.insurance_contract_manual
```

- Menyimpan data Insurance Contract Assoc Manual ke tabel historical:

```
insert into report.insurance_contract_assoc_historical select * from
staging.insurance_contract_assoc_manual
```

- Menyimpan data Insurance Contract Manual ke tabel historical:

```
insert into report.insurance_contract_historical select reporting_dt, entity_id,
insurance_contract_id, endorsement_type_cd, issue_dt, product_line_id, kob_id,
product_id, source_of_business, ceding_id, cohort_id, ceded_flg,
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
reins_prop_cover_flg, reins_treaty_flg, endorsement_dt, begin_cov_dt,
end_cov_dt, currency_cd, prem_amt, base_alloc, comm_broker_amt,
commission_amt, discount_amt, admin_fee_amt, pd, lgd,
transition_approach_cd, transition_cumulative_oci_amt, year_defisit_clause,
management_pct, profit_comm_pct, re_profit_comm_amt, re_profit_comm_dt,
source_data, category_data, null as dcno, null as dcinduk, null as dcstatus, null as
dcnofile, null as dcnotaretro, null as dcjenis, null as kacob, null as
insurance_contract_id_2, null::date as dcdateloss, null as dcclaim, null as
dcsalvage, null as dcbebanclaim, null as subcob, null as dcnoold, null::date as
dctanggal FROM staging.insurance_contract_manual
```

- Melakukan transformasi data Risk Factor

```
with risk_factor as ( delete from report.risk_factor returning risk_factor_id,
base_risk_factor_id, maturity_length_of_time, maturity_time_uom_cd,
currency_cd, risk_factor_role_cd, risk_factor_measure_type_cd,
risk_factor_category_cd, volatility_type_cd, fx_type_cd, ir_type_cd,
to_currency_cd, from_currency_cd ), usd_data as ( select a.risk_factor_id,
a.base_risk_factor_id, a.maturity_length_of_time, a.maturity_time_uom_cd,
a.currency_cd, a.risk_factor_role_cd, a.risk_factor_measure_type_cd,
a.risk_factor_category_cd, a.volatility_type_cd, a.fx_type_cd, a.ir_type_cd,
a.to_currency_cd, a.from_currency_cd from risk_factor a where risk_factor_id
like '%_USD_%' and risk_factor_id like 'IFRS17_%' ), not_usd_data as ( select
a.risk_factor_id, a.base_risk_factor_id, a.maturity_length_of_time,
a.maturity_time_uom_cd, a.currency_cd, a.risk_factor_role_cd,
a.risk_factor_measure_type_cd, a.risk_factor_category_cd, a.volatility_type_cd,
a.fx_type_cd, a.ir_type_cd, a.to_currency_cd, a.from_currency_cd from
risk_factor a left join usd_data b on a.risk_factor_id = b.risk_factor_id where
b.risk_factor_id is null ), report_risk_factor as ( select
replace(risk_factor_id,'USD',rc.code) risk_factor_id, base_risk_factor_id,
maturity_length_of_time, maturity_time_uom_cd, currency_cd,
risk_factor_role_cd, risk_factor_measure_type_cd, risk_factor_category_cd,
volatility_type_cd, fx_type_cd, ir_type_cd, to_currency_cd, from_currency_cd
from usd_data a left join (select distinct code from report.ref_currency where
code not in ('IDR','USD')) rc on 1=1 ) insert into report.risk_factor select * from
report_risk_factor union all select * from usd_data union all select * from
not_usd_data
```

- Melakukan transformasi data Risk Factor Curve

```
with risk_factor_curve as ( delete from report.risk_factor_curve returning
curve_id, curve_role_cd, currency_cd, curve_desc ), usd_data as ( select
a.curve_id, a.curve_role_cd, a.currency_cd, a.curve_desc from risk_factor_curve
a where curve_id like '%_USD_%' and curve_id like 'IFRS17_%' ), not_usd_data as
( select a.curve_id, a.curve_role_cd, a.currency_cd, a.curve_desc from
risk_factor_curve a left join usd_data b on a.curve_id = b.curve_id where
b.curve_id is null ), report_risk_factor_curve as ( select
```



1.0

TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE



23 Februari 2024

```
replace(curve_id,'USD',rc.code) curve_id, curve_role_cd, rc.code as currency_cd,
curve_desc from usd_data a left join (select distinct code from report.ref_currency
where code not in ('IDR','USD')) rc on 1=1 ) insert into report.risk_factor_curve
select * from report_risk_factor_curve union all select * from usd_data union all
select * from not_usd_data
```

- Melakukan transformasi data Risk Factor X Risk Factor Curve

```
with      risk_factor_x_risk_fctr_curve      as      (      delete      from
report.risk_factor_x_risk_fctr_curve      returning      curve_id,      risk_factor_id,
rln_risk_factor_attribute_cd ), usd_data as ( select a.curve_id, a.risk_factor_id,
a.rln_risk_factor_attribute_cd from risk_factor_x_risk_fctr_curve a where
curve_id like '%_USD_%' and risk_factor_id like 'IFRS17_%' ), not_usd_data as (
select      a.curve_id,      a.risk_factor_id,      a.rln_risk_factor_attribute_cd from
risk_factor_x_risk_fctr_curve a left join usd_data b on a.curve_id = b.curve_id
where b.curve_id is null ), report_risk_factor_x_risk_fctr_curve as ( select
replace(curve_id,'USD',rc.code) curve_id, replace(risk_factor_id,'USD',rc.code)
risk_factor_id, rln_risk_factor_attribute_cd from usd_data a left join (select
distinct code from report.ref_currency where code not in ('IDR','USD')) rc on 1=1 )
insert      into      report.risk_factor_x_risk_fctr_curve      select      *      from
report_risk_factor_x_risk_fctr_curve union all select * from usd_data union all
select * from not_usd_data
```

- Melakukan transformasi data Quotes

```
with quotes as ( delete from report.quotes returning quote_dt, quote_id,
maturity_dt, quote_rt ), usd_data as ( select a.quote_dt, a.quote_id,
a.maturity_dt, a.quote_rt from quotes a where quote_id like '%_USD_%' and
quote_id like 'IFRS17_%' ), not_usd_data as ( select a.quote_dt, a.quote_id,
a.maturity_dt, a.quote_rt from quotes a left join usd_data b on a.quote_id =
b.quote_id where b.quote_id is null ), report_quote_id as ( select quote_dt,
replace(quote_id,'USD',rc.code) quote_id, maturity_dt, quote_rt from usd_data a
left join (select distinct code from report.ref_currency where code not in
('IDR','USD')) rc on 1=1 ) insert into report.quotes select * from report_quote_id
union all select * from usd_data union all select * from not_usd_data
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

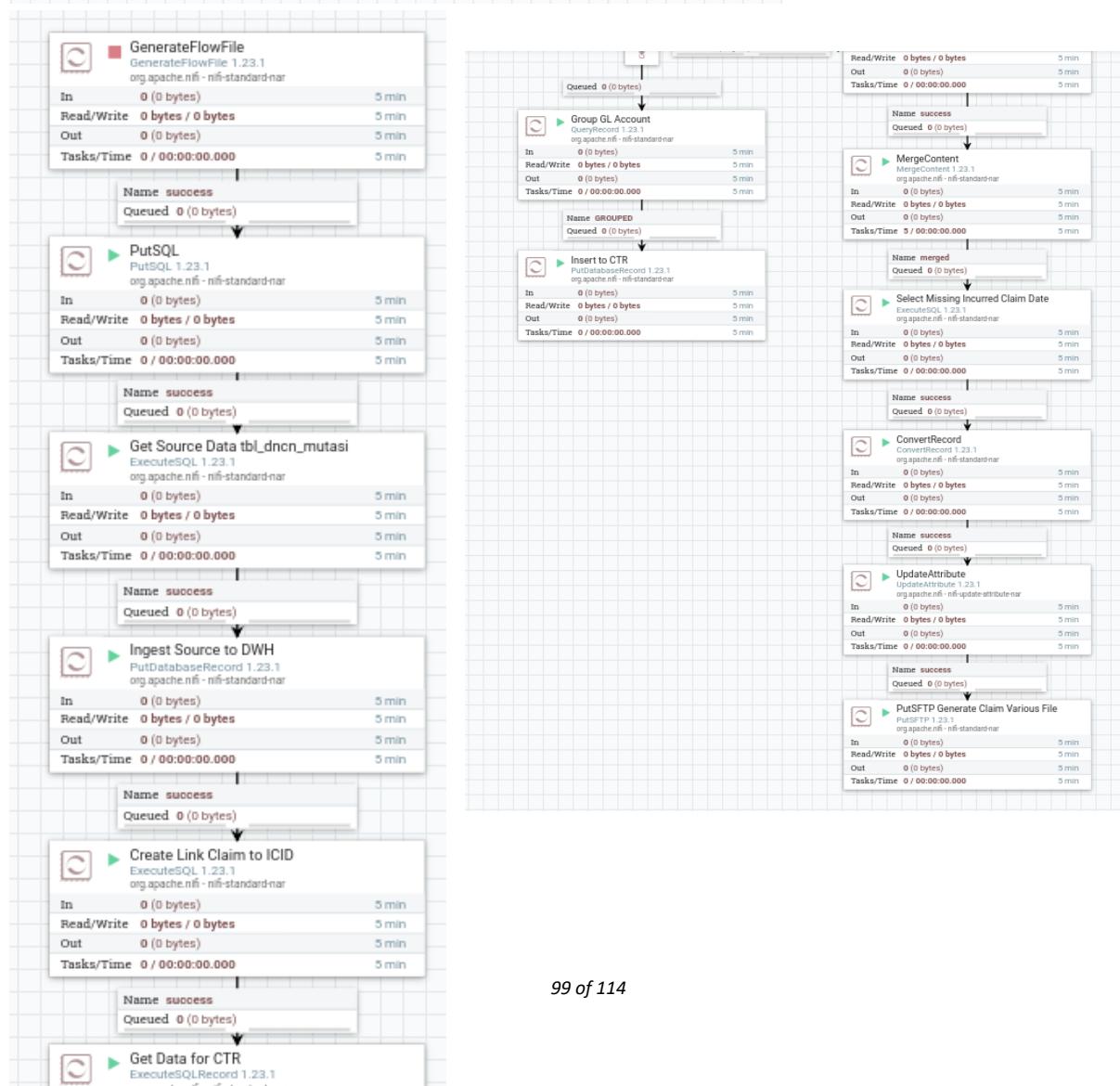
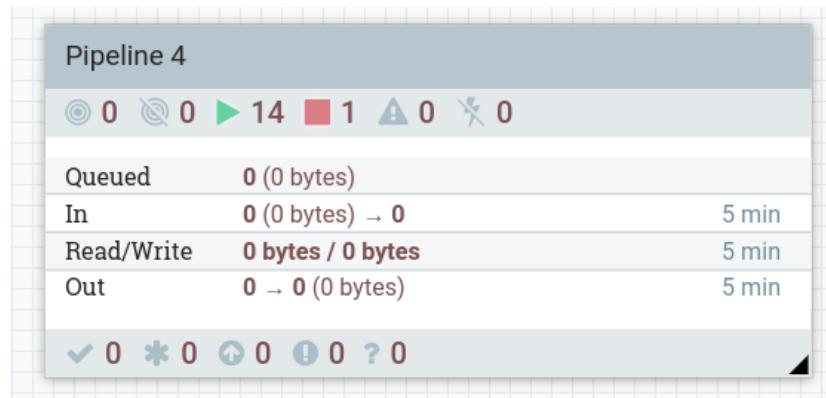
1.0



23 Februari 2024

Pipeline 4

Pada Process Group Pipeline 4 berisi Processor untuk melakukan penarikan data tabel dncn_mutasi di SQL Server Tugure, kemudian generate data CTR menggunakan data dncn_mutasi yang baru di ingest dan data insurance contract pada Process Group Pipeline 2. Data CTR selanjutnya akan disimpan di database PostgreSQL. Selain data CTR akan ada file claim_various dengan extension .csv yang mungkin di generate apabila terdapat data dengan incurred_date yang kosong. Process Group Pipeline 4 ini berjalan setiap bulan tanggal 1 pukul 01.00 WIB.





TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0

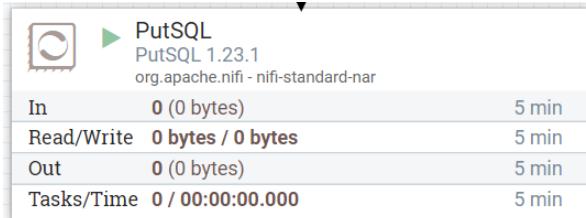


23 Februari 2024

Di dalam Process Group Pipeline 4 terdapat beberapa proses yaitu:

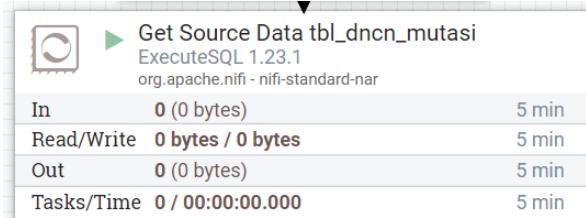
1. Proses Ingestion source data SQL Server

- PutSQL



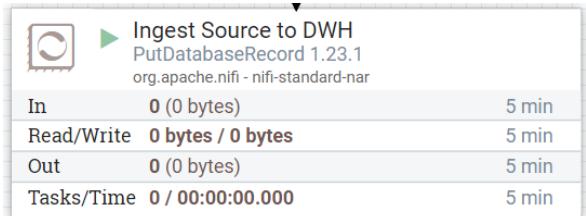
Processor ini digunakan untuk mengeksekusi perintah SQL ke database PostgreSQL DWH, perintah SQL nya untuk menghapus (truncate) data dari tabel di database PostgreSQL DWH.

- ExecuteSQL



Processor ini digunakan untuk mengeksekusi perintah SQL ke database SQL Server source data dan mengambil hasilnya kemudian hasilnya dijadikan aliran data (flowfile).

- PutDatabaseRecord



Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel di database DWH.

2. Proses Link Claim dengan Insurance Contract

- ExecuteSQL



Processor ini digunakan untuk mengeksekusi perintah SQL sebagai berikut dimana hasilnya kemudian dijadikan aliran data (flowfile).



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0

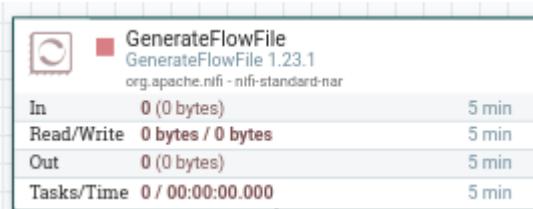


23 Februari 2024

```
with claims_mongodb as ( select distinct register_claim, transaction->>'document_number' as document_number from staging.claims c ) --select * from claims_mongodb; , fadmnf2013 as ( select nofileinduk, orc, nodebit from staging.admnf2013 where nodebit != "" ) , data_admnf2013 as ( select nofileinduk, orc, min(nodebit) nodebit, count(*) cnt from fadmnf2013 group by nofileinduk, orc ) --select * from data_admnf2013; , data_admnf2013_1 as ( select nofileinduk, orc, nodebit from data_admnf2013 where cnt = 1 ) , data_admnf2013_2 as ( select nofileinduk, orc, replace(nodebit,concat('.',split_part(nodebit,'.',4),'.'),'.00.') nodebit from data_admnf2013 where cnt > 1 ) , adm13 as ( select * from data_admnf2013_1 union all select * from data_admnf2013_2 ) , dncn_non_claim as ( select dcno, dcurr, dcnoold, dcnofile, subcob, substring(dcno, 4,3) as kdtrx from staging.tbl_dncn where left(dcno,2) in ('01', '02') ), dncn_claim as ( select dcnofile, dcno, subcob, dcurr, kdcob, dcurref, dcnoold, dcclaim, dcbebanclaim, dcsalvage, dcnet, dcdateloss, dctanggal, substring(dcno,4,3) as kdtrx from staging.tbl_dncn td where left(dcno,2) in ('03','04') ), claim_non_blips as ( select dcno, kdcob, dcurr, dcurref, dcnoold, dctanggal, dcnofile, dcdateloss, sum(dcclaim) dcclaim, sum(dcbebanclaim) dcbebanclaim, sum(dcsalvage) dcsalvage, sum(dcnet) dcnet, substring(dcno, 4,3) as kdtrx, min(nofileinduk) nofileinduk from dncn_claim join staging.admnc2000 on dcurref = noclaim where left(dcno,2) in ('03','04') and nofileinduk != 0 group by 1,2,3,4,5,6,7,8 ), link_claim_blips as ( select a.dcno as dcno_claim, c.dcno as dcno_sesi, a.dcclaim, a.dcbebanclaim, a.dcsalvage, a.dctanggal, a.dcnoold, a.dcnofile, a.dcurref, a.dcnet, a.dcdateloss from dncn_claim a join claims_mongodb b on a.dcnofile = b.register_claim join dncn_non_claim c on b.document_number = c.dcnofile and a.subcob = c.subcob and a.kdtrx = c.kdtrx and a.dcurr = c.dcurr ), link_claim_non_blips as ( select a.dcno as dcno_claim, c.dcno as dcno_sesi, a.dcclaim, a.dcbebanclaim, a.dcsalvage, a.dctanggal, a.dcnoold, a.dcnofile, a.dcurref, a.dcnet, a.dcdateloss from claim_non_blips a join adm13 b on a.nofileinduk = b.nofileinduk and a.dcurr = b.orc join dncn_non_claim c on b.nodebit = c.dcnoold and a.kdtrx = c.kdtrx ), all_link_claim as ( select dcno_sesi, dcno_claim from link_claim_blips union all select dcno_sesi, dcno_claim from link_claim_non_blips union all select dcno_sesi, dcno_claim from staging.claim_not_found where dcno_sesi is not null ) , insurance_contract as ( select reporting_dt, insurance_contract_id, dcno from report.master_icid union select reporting_dt, insurance_contract_id, dcno from report.insurance_contract_historical ) insert into staging.ref_link_claim_icid select insurance_contract_id, dcno_claim, dcno_sesi from all_link_claim a join insurance_contract b on a.dcno_sesi = b.dcno;
```

3. Proses GL Account Balance Segment CTR

- GenerateFlowFile





TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0

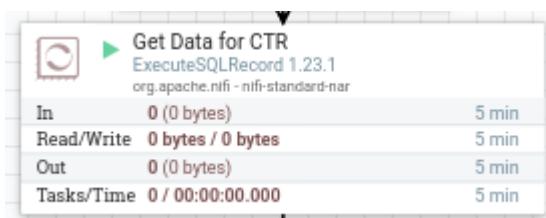


23 Februari 2024

Processor ini digunakan untuk membuat sebuah atribut yang akan digunakan pada processor-processor CTR di bawahnya. Atribut yang dihasilkan adalah sebagai berikut:

- Start_month_period
- End_month_period
- Insurance_contract_tablename
- Tablename
- Year_period

- ExecuteSQLRecord



Processor ini digunakan untuk mengeksekusi beberapa perintah SQL sebagai berikut dimana hasilnya kemudian dijadikan aliran data (flowfile):

- Pre-Query (digunakan untuk delete tabel CTR historical)

```
DELETE      FROM      report.gl_account_balance_segment_ctr_historical      WHERE
EXTRACT(YEAR FROM account_period_end_dt) = ${year_period} AND EXTRACT(MONTH
FROM account_period_end_dt) = ${end_month_period};
```

- Select Query (digunakan untuk generate data CTR)

```
WITH date_range AS (SELECT MAKE_DATE(${year_period}, ${start_month_period}, 1) AS
first_date_of_year,      (date_trunc('month',      (${year_period}::text      ||      '-'      ||
${end_month_period}::text || '-01')::date) + interval '1 month' - interval '1 day')::date AS
last_date_of_year ) --select * from date_range; , historical as ( select dcno,
insurance_contract_id, reporting_dt, kdcob, kob_id, source_of_business, ceded_flg from
report.${insurance_contract_tablename} cross join date_range where reporting_dt <=
last_date_of_year ) --select * from historical; , dncn as (select td.dcno, td.dctanggal, case
when td.kdcob = '' or td.kdcob is null then tdd.kdcob else td.kdcob, dcurr,
coalesce(dcdateloss, cv.incurred_claim_dt) as dcdateloss, coalesce(dcgrosspremium,0) as
gross_prem, coalesce(dccommission,0) as comm, coalesce(dcbrokerage,0) as broker,
coalesce(dcclaim,0) as claim, coalesce(dcbebanclaim,0) as beban_claim,
coalesce(dcsalvage,0) as salvage, case when td.dcno like '%-U%' then 'WebUpload' when
td.dcno like '%-C%' and td.dcno not like '%-U%' then 'WebCredit' else 'BlipsOps' end as
category from staging.tbl_dncn td left join staging.tbl_detaildncn tdd on td.dcno =
tdd.dcno left join staging.claim_various cv on td.dcno = cv.dcno ) --select * from dncn; ,
mutasi as ( select id, dcno, ccy, dk, isposting, paydk * -1 as paydk, pay, jnsket, subjns,
novoucher, tglvoucher, tglefektif, case when dcno like '%-U%' then 'WebUpload' when
dcno like '%-C%' and dcno not like '%-U%' then 'WebCredit' else 'BlipsOps' end as category
from staging.tbl_dncn_mutasi tdm where isposting = '1' ), temp as ( select distinct id,
dncn.dcno, kdcob, jnsket, dctanggal, subjns, tglefektif, tglvoucher, novoucher,
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
(date_trunc('MONTH', case when jnsket = 'ADJUST' and (novoucher like '%BS%' or novoucher like '%RS%' or novoucher like '%RV%' or novoucher like '%AD%') then tglvoucher when jnsket = 'ADJUST' and novoucher like '%AN%' then (case when tglefektif is null or tglefektif < dctanggal then tglvoucher else tglefektif end) else tglefektif end) + INTERVAL '1 month - 1 day')::date as tf_tglefektif, dcdateloss, coalesce(dccurr, ccy) as ccy, gross_prem, comm, broker, claim, beban_claim, salvage, dncn.category, paydk, pay, dk from dncn left join mutasi on dncn.dcno = mutasi.dcno ) --select * from temp; , nota_paid_sesi as ( select distinct id, insurance_contract_id, temp.subjns, temp.dcno, tf_tglefektif, temp.dctanggal, null::date as tf_dctanggal, ich.kob_id, ich.source_of_business, temp.dcdateloss, null::date as tf_dcdateloss, ceded_flg, temp.kdcob, category, ccy, dk, coalesce(pay,0) as pay, coalesce(paydk,0) as paydk, gross_prem, comm, broker, claim, beban_claim, salvage, first_date_of_year, last_date_of_year from temp cross join date_range join historical ich on temp.dcno = ich.dcno and temp.kdcob = ich.kdcob where left(temp.dcno, 2) in ('01', '02') and pay is not null ) --select * from nota_paid_sesi; , nota_claim as ( select distinct id, rlci.insurance_contract_id, temp.subjns, temp.dcno, tf_tglefektif, temp.dctanggal, (date_trunc('MONTH', temp.dctanggal)::date) + INTERVAL '1 month - 1 day')::date as tf_dctanggal, ich.kob_id, ich.source_of_business, temp.dcdateloss, CASE WHEN extract(YEAR FROM temp.dcdateloss) < extract(YEAR FROM last_date_of_year) THEN ((extract(YEAR FROM last_date_of_year) - 1) || '-12-31')::date ELSE last_date_of_year END as tf_dcdateloss, ich.ceded_flg, temp.kdcob, category, ccy, dk, pay, paydk, gross_prem, comm, broker, claim, beban_claim, salvage, first_date_of_year, last_date_of_year from temp cross join date_range left join staging.ref_link_claim_icid rlci on temp.dcno = rlci.dcno_claim left join historical ich on rlci.insurance_contract_id = ich.insurance_contract_id and temp.kdcob = ich.kdcob WHERE left(temp.dcno, 2) in ('03','04') and rlci.insurance_contract_id is not null ) --select * from nota_claim; , raw as ( select *, 'non_claim' as nota_type from nota_paid_sesi where tf_tglefektif between first_date_of_year and last_date_of_year union select *, 'claim_reported' as nota_type from nota_claim where tf_dctanggal <= last_date_of_year ) select * from raw where kdcob in ('MH', 'FR', 'MV', 'EN') and kob_id = 'FC';
```

- QueryRecord

Create GL Account ID		Group GL Account	
QueryRecord 1.23.1		QueryRecord 1.23.1	
org.apache.nifi - nifi-standard-nar		org.apache.nifi - nifi-standard-nar	
In	0 (0 bytes)	5 min	5 min
Read/Write	0 bytes / 0 bytes	5 min	5 min
Out	0 (0 bytes)	5 min	5 min
Tasks/Time	0 / 00:00:00.000	5 min	5 min

Processor ini digunakan untuk melakukan perintah SQL terhadap flowfile yang dihasilkan sebelumnya. Adapun beberapa perintah SQL yang dijalankan adalah untuk membuat dan memisahkan flowfile per gl_account_id nya masing-masing, detailnya adalah sebagai berikut:

- Claims_payment



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
SELECT 'TRE' AS entity_id, 'LCL_ACTUAL CLAIMS PAYMENT' AS gl_account_id, insurance_contract_id, tf_tglefektif AS cashflow_dt, ceded_flg, dcdateloss AS incurred_claim_dt, ccy AS currency_cd, last_date_of_year AS account_period_end_dt, paydk AS account_period_closing_bal_amt, dcno, category AS source_data, dk, pay, dcdateloss, dctanggal FROM FLOWFILE WHERE nota_type = 'claim_reported' AND tf_tglefektif BETWEEN first_date_of_year AND last_date_of_year AND paydk IS NOT NULL AND subjns = 1
```

- Claims_reported

```
SELECT 'TRE' AS entity_id, 'LCL_ACTUAL CLAIMS REPORTED' AS gl_account_id, insurance_contract_id, tf_dctanggal AS cashflow_dt, ceded_flg, dcdateloss AS incurred_claim_dt, ccy AS currency_cd, last_date_of_year AS account_period_end_dt, CASE WHEN ABS(claim) > 0 THEN ABS(claim) WHEN ABS(beban_claim) > 0 THEN ABS(beban_claim) ELSE ABS(salvage) END * CASE WHEN dcno LIKE '03%' THEN 1 ELSE -1 END AS account_period_closing_bal_amt, dcno, category AS source_data, dk, pay, dcdateloss, dctanggal FROM FLOWFILE WHERE nota_type = 'claim_reported' AND tf_dctanggal BETWEEN first_date_of_year AND last_date_of_year AND subjns = 1
```

- Commission

```
SELECT 'TRE' AS entity_id, CASE WHEN ceded_flg = 'Y' THEN 'LCL_ACTUAL REINS COM' ELSE 'LCL_ACTUAL COMM' END AS gl_account_id, insurance_contract_id, tf_tglefektif AS cashflow_dt, ceded_flg, '1960-01-01' AS incurred_claim_dt, ccy AS currency_cd, last_date_of_year AS account_period_end_dt, paydk - (paydk / (1 - ABS((comm+broker)/gross_prem))) AS account_period_closing_bal_amt, dcno, category AS source_data, dk, pay, dcdateloss, dctanggal FROM FLOWFILE WHERE nota_type = 'non_claim' AND subjns = 1
```

- Other_costs

```
SELECT 'TRE' AS entity_id, 'LCL_ACTUAL OTHER COSTS' AS gl_account_id, insurance_contract_id, tf_tglefektif AS cashflow_dt, ceded_flg, '1960-01-01' AS incurred_claim_dt, ccy AS currency_cd, last_date_of_year AS account_period_end_dt, paydk AS account_period_closing_bal_amt, dcno, category AS source_data, dk, pay, dcdateloss, dctanggal FROM FLOWFILE WHERE subjns in (2,3)
```

- Actual_Premiums

```
SELECT 'TRE' AS entity_id, 'LCL_ACTUAL PREMIUMS' AS gl_account_id, insurance_contract_id, tf_tglefektif AS cashflow_dt, ceded_flg, '1960-01-01' AS incurred_claim_dt, ccy AS currency_cd, last_date_of_year AS account_period_end_dt, paydk / (1 - ABS((comm+broker)/gross_prem)) AS account_period_closing_bal_amt, dcno, category AS source_data, dk, pay, dcdateloss, dctanggal FROM FLOWFILE WHERE nota_type = 'non_claim' AND subjns = 1
```

- RBNP_closing

```

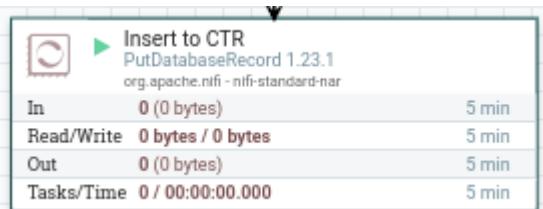
SELECT 'TRE' AS entity_id, 'LCL_ACTUAL_RBNP_CLOSING' AS gl_account_id,
insurance_contract_id, last_date_of_year AS cashflow_dt, ceded_flg, tf_dcdateloss
AS incurred_claim_dt, tf_tglefektif, ccy AS currency_cd, last_date_of_year AS
account_period_end_dt, CASE WHEN tf_tglefektif BETWEEN first_date_of_year AND
last_date_of_year AND ABS(claim) > ABS(COALESCE(pay, 0)) THEN
ABS(COALESCE(pay, 0))-ABS(claim) WHEN tf_tglefektif BETWEEN first_date_of_year
AND last_date_of_year AND ABS(beban_claim) > ABS(COALESCE(pay, 0)) THEN
ABS(COALESCE(pay, 0))-ABS(beban_claim) WHEN tf_tglefektif BETWEEN
first_date_of_year AND last_date_of_year AND ABS(salvage) > ABS(COALESCE(pay,
0)) THEN ABS(COALESCE(pay, 0))-ABS(salvage) ELSE CASE WHEN ABS(claim) > 0 THEN
ABS(claim) WHEN ABS(beban_claim) > 0 THEN ABS(beban_claim) ELSE ABS(salvage)
END END * CASE WHEN dcno LIKE '03%' THEN 1 ELSE -1 END AS
account_period_closing_bal_amt, dcno, category AS source_data, dk, pay,
dcdateloss, dctanggal FROM flowfile WHERE nota_type = 'claim_reported' AND
tf_dcdateloss <= last_date_of_year and (tf_tglefektif > last_date_of_year or
abs(claim)+abs(beban_claim)+abs(salvage) > abs(coalesce(pay,0))) AND subjns = 1
  
```

- Grouping gl_account transactions

```

SELECT entity_id, gl_account_id, insurance_contract_id, cashflow_dt, ceded_flg,
incurred_claim_dt, currency_cd, account_period_end_dt,
SUM(account_period_closing_bal_amt) AS account_period_closing_bal_amt FROM
FLOWFILE GROUP BY entity_id, gl_account_id, insurance_contract_id, cashflow_dt,
ceded_flg, incurred_claim_dt, currency_cd, account_period_end_dt
  
```

- PutDatabaseRecord



Processor ini digunakan untuk melakukan input data ke tabel terkait di database.

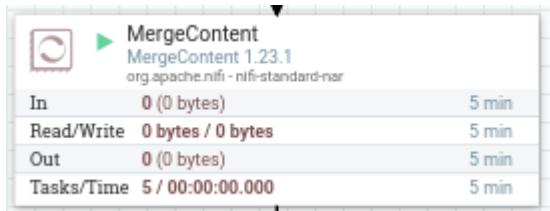
4. Proses File Claim Various

- PutDatabaseRecord



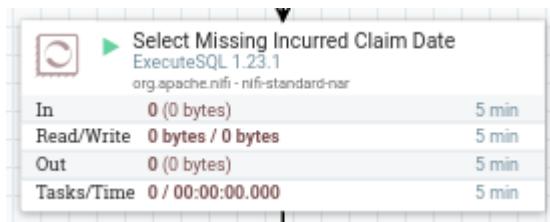
Processor ini digunakan untuk melakukan input data ke tabel terkait di database.

- MergeContent



Processor ini digunakan untuk menggabungkan beberapa flowfile menjadi satu supaya processor selanjutnya hanya menerima 1 flowfile.

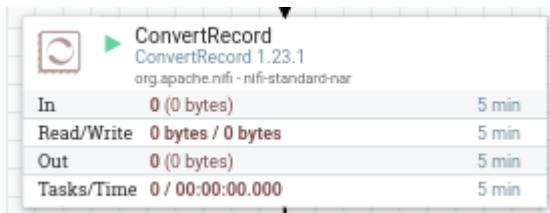
- ExecuteSQL



Processor ini digunakan untuk mengeksekusi perintah SQL dimana perintah yang dijalankan adalah untuk mencari data dari tabel CTR yang memiliki incurred_claim_dt null. Detail SQL nya adalah sebagai berikut:

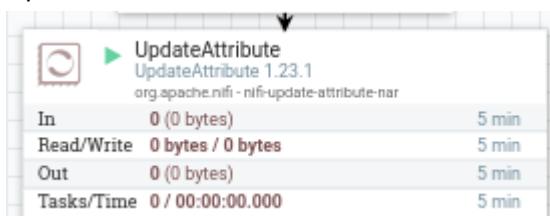
```
select distinct dcno, incurred_claim_dt from report.${tablename}_tgr where incurred_claim_dt is null
```

- ConvertRecord



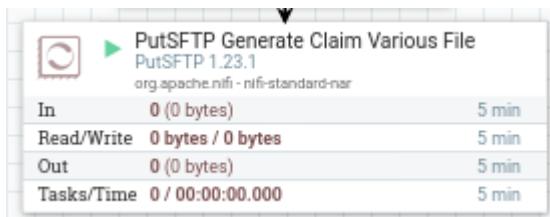
Processor ini digunakan untuk mengubah output flowfile dengan format tertentu menjadi format yang berbeda. Pada process group ini, ConvertRecord digunakan untuk mengubah flowfile dari Avro menjadi CSV.

- UpdateAttribute



Processor ini digunakan untuk menambahkan atau mengubah attribute pada suatu flowfile, pada process group ini UpdateAttribute digunakan untuk mengubah attribute filename pada NiFi.

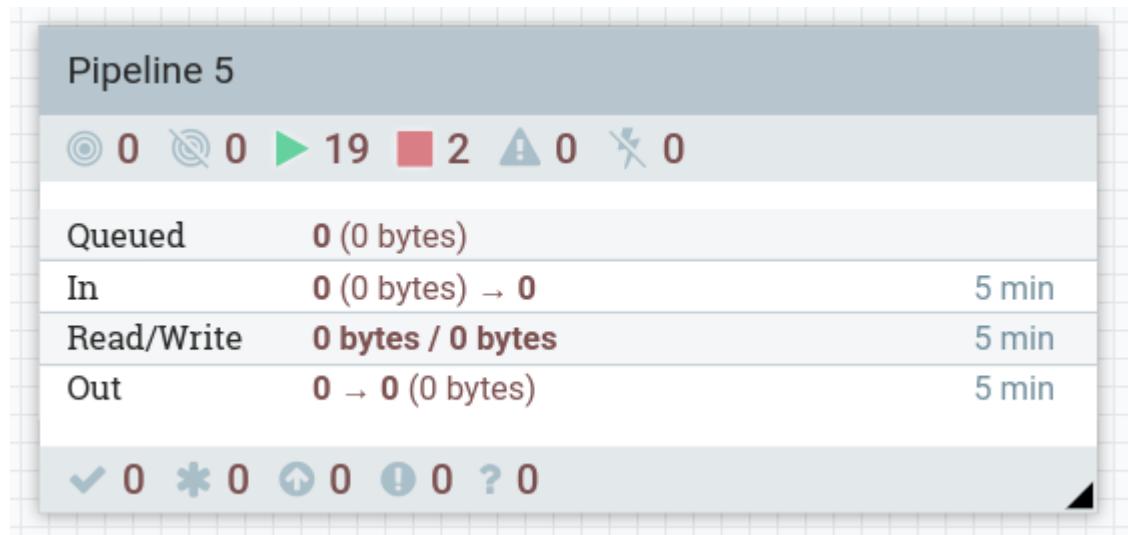
– PutSFTP



Processor ini digunakan untuk mengunggah (upload) file dari aliran data (flowfile) ke sebuah server SFTP (Secure File Transfer Protocol).

Pipeline 5

Pada Process Group Pipeline 5 berisi Processor untuk melakukan ingestion data hasil output SAS yang memiliki file extension .txt. Hasil ingestion tersebut akan disimpan di database PostgreSQL dan berjalan setiap bulan tanggal 1 – 20 pukul 01.00 WIB.



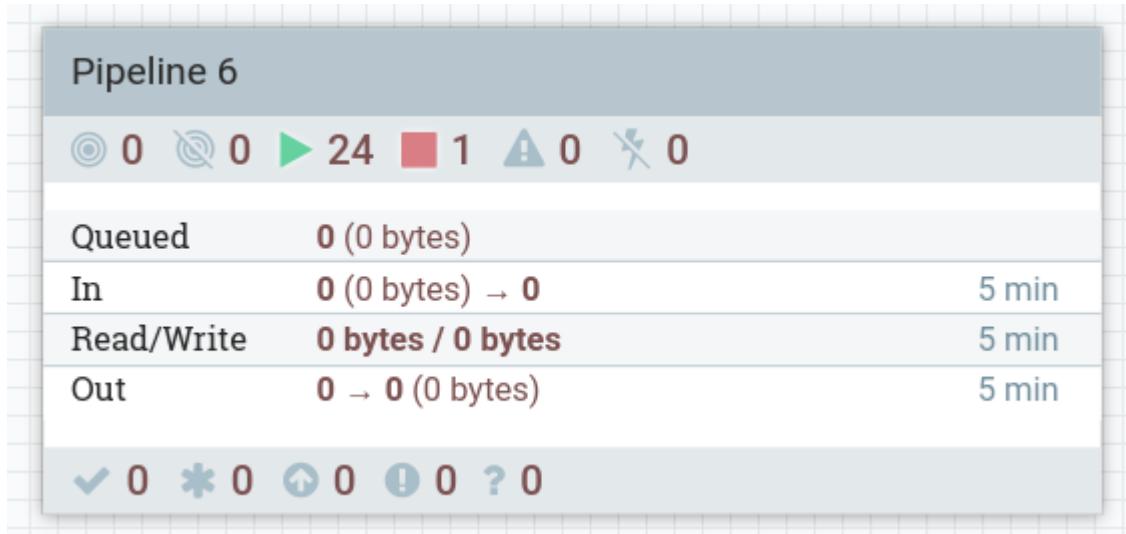
Proses-proses yang terdapat dalam Pipeline 5 ini adalah:

1. Proses Ingestion File tbl_jurnal.txt dan tbl_jurnaldetail.txt

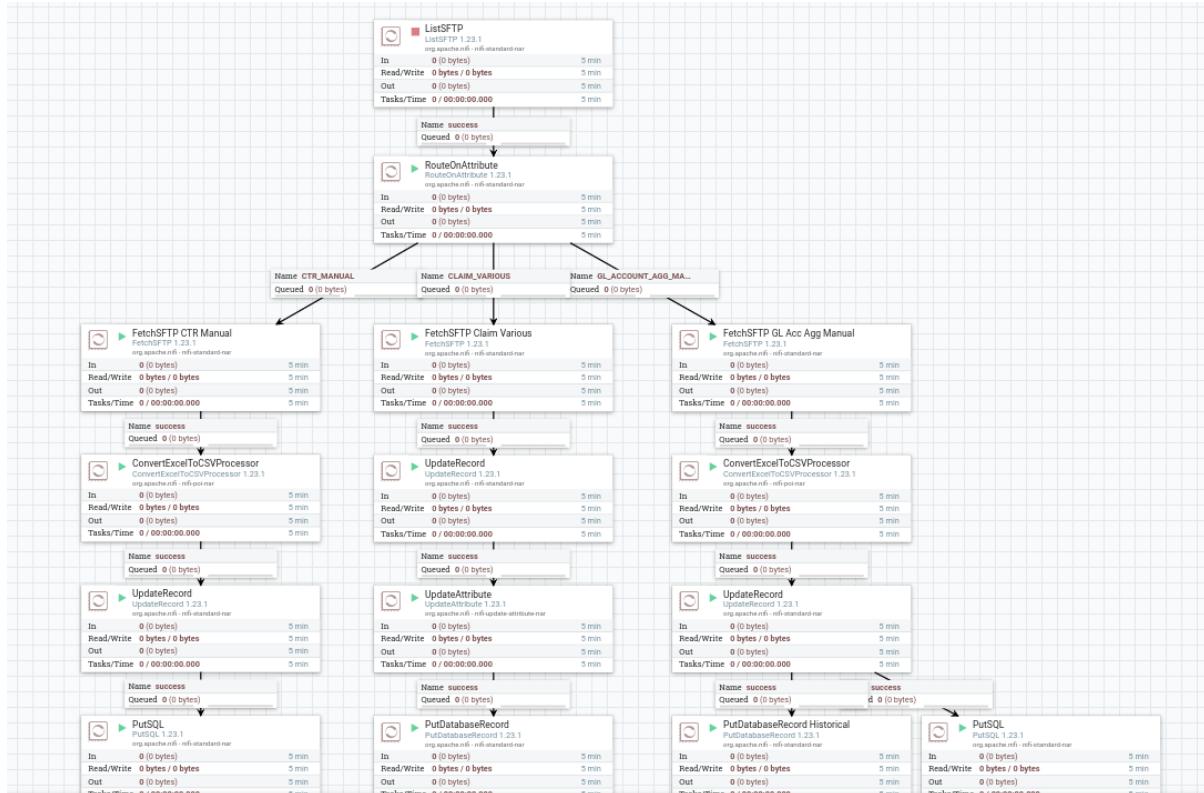
–

Pipeline 6

Pada Process Group Pipeline 6 berisi Processor untuk melakukan ingestion terhadap beberapa file yaitu claim_various, gl_account_balance_segment_ctr_manual dan juga gl_account_balance_agg_manual. Data hasil ingestion tersebut selanjutnya akan disimpan di database PostgreSQL dan melakukan update terhadap tabel CTR yang dijalankan pada Process Group Pipeline 4. Process Group Pipeline 6 sendiri akan berjalan setiap bulan tanggal 2 – 5 pukul 01.00 WIB.

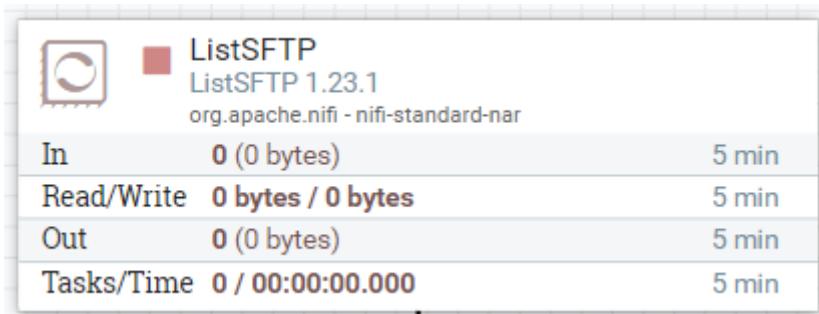


Proses-proses yang terdapat dalam Pipeline 6 adalah:



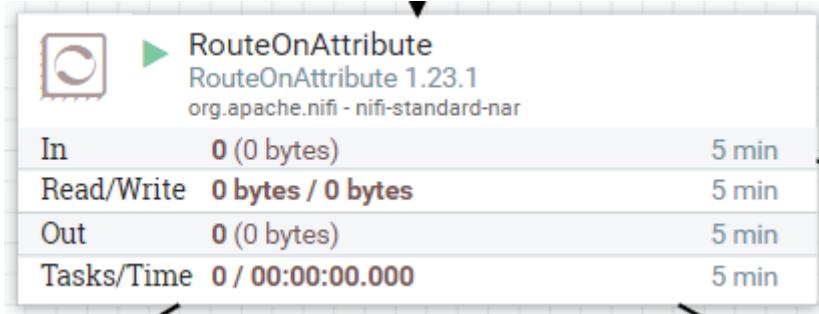
1. Proses Ingestion File CTR Manual, CTR Agg Manual, dan Claim Various

- ListSFTP



Processor ini digunakan untuk mendapatkan daftar file dari sebuah server SFTP (Secure File Transfer Protocol), untuk mendapatkan file claim_various, gl_account_balance_segment_ctr_manual, dan juga gl_account_balance_segment_agg

- RouteOnAttribute

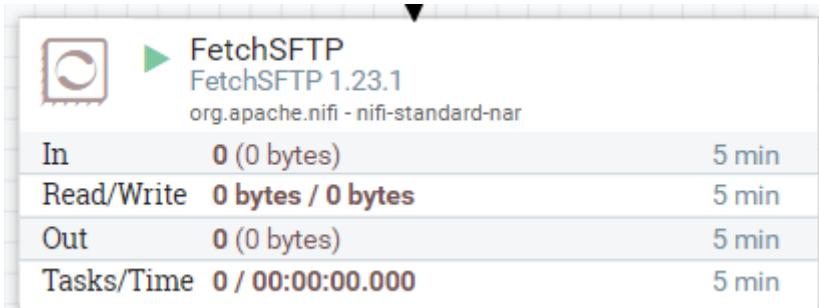


Processor ini digunakan untuk membagi aliran data (flowfile) menjadi cabang-cabang yang berbeda berdasarkan nilai atribut dari setiap FlowFile.

SETTINGS	SCHEDULING	PROPERTIES	RELATIONSHIPS	COMMENTS
Required field				
Property				Value
Routing Strategy	?	Route to Property name		
CLAIM_VARIOUS	?	<code> \${filename:contains('claim_various')}</code>		
CTR_MANUAL	?	<code> \${filename:contains('gl_account_balance_segment_ctr')}</code>		
GL_ACCOUNT_AGG_MANUAL	?	<code> \${filename:contains('gl_account_balance_segment_ag...')}</code>		

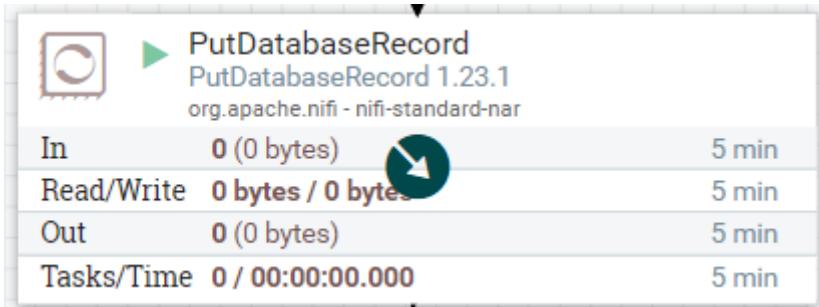
disini processor membagi flowfile menjadi 3 yaitu CLAIM_VARIOUS, CTR_MANUAL dan GL_ACCOUNT_AGG_MANUAL.

- FetchSFTP



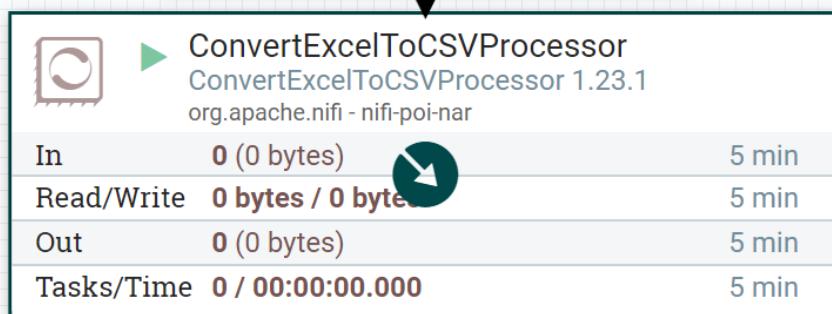
Processor ini digunakan untuk mengunduh (fetch) file dari server SFTP (Secure File Transfer Protocol), mengambil datanya kemudian datanya dijadikan aliran data (flowfile).

- PutDatabaseRecord



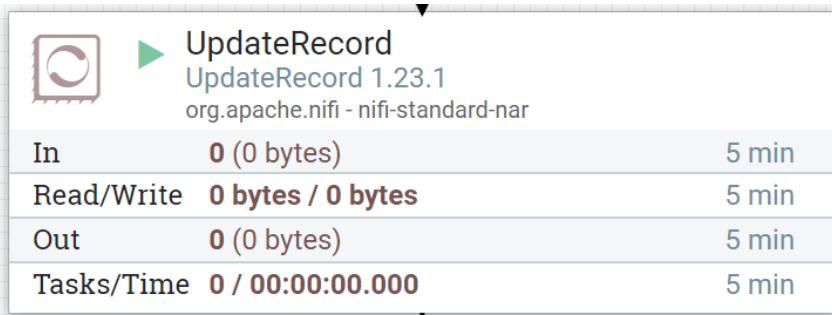
Processor ini digunakan untuk menyimpan data dari aliran data (flowfile) ke dalam tabel di database PostgreSQL DWH.

- ConvertExcelToCSVProcessor



Processor ini digunakan untuk melakukan konversi terhadap aliran data (flowfile) yang berbentuk excel ke format csv agar lebih mudah diproses oleh processor Nifi.

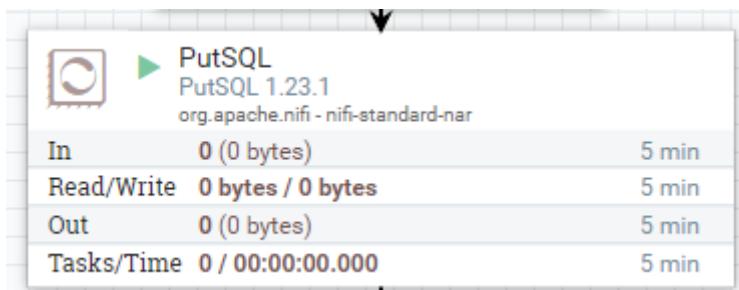
- UpdateRecord



Processor ini digunakan untuk memperbarui atau menambahkan record pada setiap FlowFile yang melewati prosesor tersebut dalam aliran data (flowfile). Didalam processor ini attribut yang diperbarui atau ditambahkan adalah sebagai berikut:

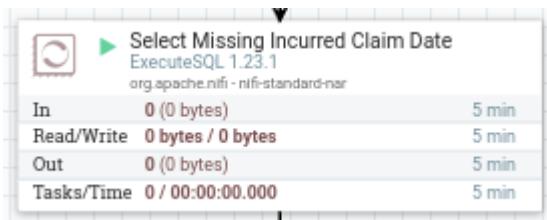
- ctr_manual:
 - Mengubah format tanggal account_period_end_dt, cashflow_dt, incurred_claim_dt dari yyyyymmdd menjadi yyyy-mm-dd
- claim_various:
 - Mengubah format tanggal incurred_claim_dt dari yyyyymmdd menjadi yyyy-mm-dd
- GI_acc_agg_manual:
 - Mengubah format tanggal account_period_start_dt, account_period_end_dt, cashflow_dt, dan incurred_claim_dt dari yyyyymmdd menjadi yyyy-mm-dd

- PutSQL



Processor ini digunakan untuk mengeksekusi perintah SQL ke database PostgreSQL DWH, perintah SQL nya untuk menghapus (truncate) data dari tabel di database PostgreSQL DWH.

- ExecuteSQL



Processor ini digunakan untuk mengeksekusi perintah SQL dimana perintah yang dijalankan adalah untuk membuat kembali tabel CTR untuk kemudian digabungkan dengan CTR Manualnya apabila ada file claim_various, jika tidak ada claim_various maka hanya akan menambahkan data CTR Manual ke tabel CTR. Detail SQL nya adalah sebagai berikut:

- Select Query pada CTR Manual

```
insert into report.gl_account_balance_segment_ctr select entity_id, gl_account_id,
insurance_contract_id, cashflow_dt, ceded_flg, incurred_claim_dt, currency_cd,
account_period_end_dt, account_period_closing_bal_amt from
staging.gl_account_balance_segment_ctr_manual
```

- Post-Query pada CTR Manual

```
insert into report.gl_account_balance_segment_ctr_historical select entity_id,
gl_account_id, insurance_contract_id, cashflow_dt, ceded_flg, incurred_claim_dt,
currency_cd, account_period_end_dt, account_period_closing_bal_amt from
staging.gl_account_balance_segment_ctr_manual
```

- Query Link Claim Insurance Contract pada Claim Various

```
with claims_mongodb as ( select distinct register_claim, transaction-
->>'document_number' as document_number from staging.claims c ) --select * from
claims_mongodb; , fadmnf2013 as ( select nofileinduk, orc, nodebit from
staging.admnf2013 where nodebit != "" ), data_admnf2013 as ( select nofileinduk, orc,
min(nodebit) nodebit, count(*) cnt from fadmnf2013 group by nofileinduk, orc ) --select *
from data_admnf2013; , data_admnf2013_1 as ( select nofileinduk, orc, nodebit from
data_admnf2013 where cnt = 1 ) , data_admnf2013_2 as ( select nofileinduk, orc,
replace(nodebit,concat('.',split_part(nodebit,'.',4),'.').00.')) nodebit from
data_admnf2013 where cnt > 1 ) , adm13 as ( select * from data_admnf2013_1 union all
select * from data_admnf2013_2 ) , dncn_non_claim as ( select dcno, dccurr, dcnoold,
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

```
dcnofile, subcob, substring(dcno, 4,3) as kdtrx from staging.tbl_dncn where left(dcno,2) in ('01', '02') , dncn_claim as ( select dcnofile, dcno, subcob, dccurr, kdcob, dcourref, dcnoold, dcclaim, dcbebanclaim, dcsalvage, dcnet, dcdateloss, dctanggal, substring(dcno,4,3) as kdtrx from staging.tbl_dncn td where left(dcno,2) in ('03','04') ) , claim_non_blips as ( select dcno, kdcob, dccurr, dcourref, dcnoold, dctanggal, dcnofile, dcdateloss, sum(dcclaim) dcclaim, sum(dcbebanclaim) dcbebanclaim, sum(dcsalvage) dcsalvage, sum(dcnet) dcnet, substring(dcno, 4,3) as kdtrx, min(nofileinduk) nofileinduk from dncn_claim join staging.admnc2000 on dcourref = noclaim where left(dcno,2) in ('03','04') and nofileinduk != 0 group by 1,2,3,4,5,6,7,8 ), link_claim_blips as ( select a.dcno as dcno_claim, c.dcno as dcno_sesi, a.dcclaim, a.dcbebanclaim, a.dcsalvage, a.dctanggal, a.dcnoold, a.dcnofile, a.dcourref, a.dcnet, a.dcdateloss from dncn_claim a join claims_mongodb b on a.dcnofile = b.register_claim join dncn_non_claim c on b.document_number = c.dcnofile and a.subcob = c.subcob and a.kdtrx = c.kdtrx and a.dccurr = c.dccurr ), link_claim_non_blips as ( select a.dcno as dcno_claim, c.dcno as dcno_sesi, a.dcclaim, a.dcbebanclaim, a.dcsalvage, a.dctanggal, a.dcnoold, a.dcnofile, a.dcourref, a.dcnet, a.dcdateloss from claim_non_blips a join adm13 b on a.nofileinduk = b.nofileinduk and a.dccurr = b.orc join dncn_non_claim c on b.nodebit = c.dcnoold and a.kdtrx = c.kdtrx ), all_link_claim as ( select dcno_sesi, dcno_claim from link_claim_blips union all select dcno_sesi, dcno_claim from link_claim_non_blips union all select dcno_sesi, dcno_claim from staging.claim_not_found where dcno_sesi is not null ) , insurance_contract as ( select reporting_dt, insurance_contract_id, dcno from report.master_icid union select reporting_dt, insurance_contract_id, dcno from report.insurance_contract_historical ) insert into staging.ref_link_claim_icid select insurance_contract_id, dcno_claim, dcno_sesi from all_link_claim a join insurance_contract b on a.dcno_sesi = b.dcno;
```

- Query Get Data CTR pada Claim Various

```
WITH date_range AS ( SELECT MAKE_DATE(${year_period}, ${start_month_period}, 1) AS first_date_of_year, (date_trunc('month', (${year_period}::text || '-' || ${end_month_period}::text || '-01')::date) + interval '1 month' - interval '1 day')::date AS last_date_of_year ) --select * from date_range; , historical as ( select dcno, insurance_contract_id, reporting_dt, kdcob, kob_id, source_of_business, ceded_flg from report.${insurance_contract_tablename} cross join date_range where reporting_dt <= last_date_of_year ) --select * from historical; , dncn as ( select td.dcno, td.dctanggal, case when td.kdcob = " or td.kdcob is null then tdd.kdcob else td.kdcob end as kdcob, dccurr, coalesce(dcdateloss, cv.inurred_claim_dt) as dcdateloss, coalesce(dcgrosspremium,0) as gross_prem, coalesce(dccommission,0) as comm, coalesce(dcbrokerage,0) as broker, coalesce(dcclaim,0) as claim, coalesce(dcbebanclaim,0) as beban_claim, coalesce(dcsalvage,0) as salvage, case when td.dcno like '%-%U%' then 'WebUpload' when td.dcno like '%-%C%' and td.dcno not like '%-%U%' then 'WebCredit' else 'BlipsOps' end as category from staging.tbl_dncn td left join staging.tbl_detaildncn tdd on td.dcno = tdd.dcno left join staging.claim_various cv on td.dcno = cv.dcno ) --select * from dncn; , mutasi as ( select id, dcno, ccy, dk, isposting, paydk * -1 as paydk, pay, jnsket, subjns, novoucher, tglvoucher, tglefektif, case when dcno like '%-%U%' then 'WebUpload' when dcno
```



TECHNICAL DOCUMENT

TUGURE DATA WAREHOUSE

1.0



23 Februari 2024

like '%C%' and dcno not like '%U%' then 'WebCredit' else 'BlipsOps' end as category from staging.tbl_dncn_mutasi tdm where isposting = '1'), temp as (select distinct id, dncn.dcno, kdcob, jnsket, dctanggal, subjns, tglefektif, tglvoucher, novoucher, (date_trunc('MONTH', case when jnsket = 'ADJUST' and (novoucher like '%BS%' or novoucher like '%RS%' or novoucher like '%RV%' or novoucher like '%AD%') then tglvoucher when jnsket = 'ADJUST' and novoucher like '%AN%' then (case when tglefektif is null or tglefektif < dctanggal then tglvoucher else tglefektif end) else tglefektif end) + INTERVAL '1 month - 1 day')::date as tf_tglefektif, dcdateloss, coalesce(dccurr, ccy) as ccy, gross_prem, comm, broker, claim, beban_claim, salvage, dncn.category, paydk, pay, dk from dncn left join mutasi on dncn.dcno = mutasi.dcno) --select * from temp; , nota_paid_sesi as (select distinct id, insurance_contract_id, temp.subjns, temp.dcno, tf_tglefektif, temp.dctanggal, null::date as tf_dctanggal, ich.kob_id, ich.source_of_business, temp.dcdateloss, null::date as tf_dcdateloss, ceded_flg, temp.kdcob, category, ccy, dk, coalesce(pay,0) as pay, coalesce(paydk,0) as paydk, gross_prem, comm, broker, claim, beban_claim, salvage, first_date_of_year, last_date_of_year from temp cross join date_range join historical ich on temp.dcno = ich.dcno and temp.kdcob = ich.kdcob where left(temp.dcno, 2) in ('01', '02') and pay is not null) --select * from nota_paid_sesi; , nota_claim as (select distinct id, rlci.insurance_contract_id, temp.subjns, temp.dcno, tf_tglefektif, temp.dctanggal, (date_trunc('MONTH', temp.dctanggal)::date) + INTERVAL '1 month - 1 day')::date as tf_dctanggal, ich.kob_id, ich.source_of_business, temp.dcdateloss, CASE WHEN extract(YEAR FROM temp.dcdateloss) < extract(YEAR FROM last_date_of_year) THEN ((extract(YEAR FROM last_date_of_year) - 1) || '-12-31')::date ELSE last_date_of_year END as tf_dcdateloss, ich.ceded_flg, temp.kdcob, category, ccy, dk, pay, paydk, gross_prem, comm, broker, claim, beban_claim, salvage, first_date_of_year, last_date_of_year from temp cross join date_range left join staging.ref_link_claim_icid rlci on temp.dcno = rlci.dcno_claim left join historical ich on rlci.insurance_contract_id = ich.insurance_contract_id and temp.kdcob = ich.kdcob WHERE left(temp.dcno, 2) in ('03', '04') and rlci.insurance_contract_id is not null) --select * from nota_claim; , raw as (select *, 'non_claim' as nota_type from nota_paid_sesi where tf_tglefektif between first_date_of_year and last_date_of_year union select *, 'claim_reported' as nota_type from nota_claim where tf_dctanggal <= last_date_of_year) select * from raw where kdcob in ('MH', 'FR', 'MV', 'EN') and kob_id = 'FC';

- Query Grouping CTR pada Claim Various

```
SELECT entity_id, gl_account_id, insurance_contract_id, cashflow_dt, ceded_flg, incurred_claim_dt, currency_cd, account_period_end_dt, SUM(account_period_closing_bal_amt) AS account_period_closing_bal_amt FROM FLOWFILE GROUP BY entity_id, gl_account_id, insurance_contract_id, cashflow_dt, ceded_flg, incurred_claim_dt, currency_cd, account_period_end_dt
```