Seminar

July 25th, 2020

DATA SCIENCE INDONESIA

Data Science Indonesia
East Java Region

JATIM CAMP #5

Build Data Ecosystem for Better Analytics

Data Pipelining for Creating Great Data Ecosystem

with

Rendy Bambang Junior
Senior Data Manager, Ruangguru

Muhammad Iqbal Tawakal
Data Scientist, Gojek

Supported by

holistics

Enjoyable Data Pipeline

Rendy B. Junior

Seminar DSI Jatim, 25 Juli 2020

About

Rendy Bambang Junior

Twitter, Medium: @rendybjunior

2019 - now : Senior Data Manager at Ruangguru

2014 - 2019 : Data Engineering Lead, Data System Architect at Traveloka

Google Developer Expert - Cloud (Big Data Specialization)

Let’s start with **why**,
why we need a **data pipeline**?

Common Issues on Using Data

Common Issues to Use Data

- **Data in many places**: Need to collect data into one place (MySQL, Postgres, log files, CSV, etc.)
- **Many joins**: Need to join data from several places, including asking what the data definition is
- **Dirty data**: testing account, different format across app version, duplicates
- **Don’t know which data should we use**: payment data or transaction data?
- **Unclear definition**: what is a customer?
- **Duplicate effort on processing**: I clean and join that data, you join that too?

Examples of Dirty Data

- Pattern does not match, e.g. email “asep@gmail”, missing “.com”
- Has weird characters, e.g. phone number “081-234-5-67-89”
- Outlier unexpected values, e.g. sales date 1900-01-01
- Need to be parsed, e.g. tags “023-JABAR-022” → str.split(“-”)[1] → “JABAR”
- Duplicates of data, e.g. retries, only the last is success

Data pipeline, as **foundation**
of Data Ecosystem,
“tries” to solve those problems

Data pipeline “tries” to create

1. **One environment**: collect data from various sources into one place

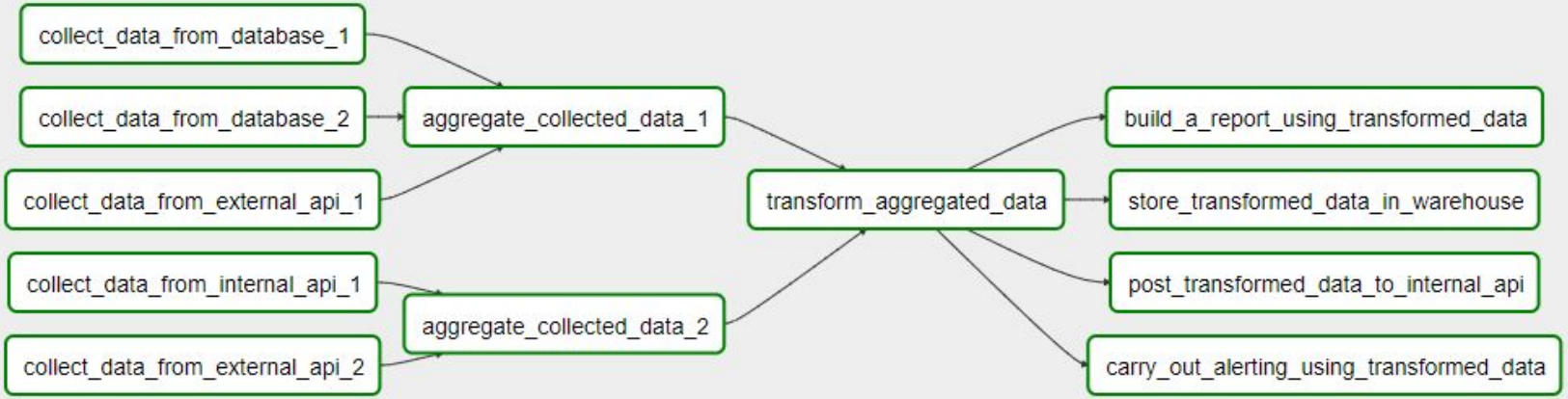
Airflow DAGs Data Profiling Browse Admin Docs About

On DAG: example_dag

Graph View | Tree View | Task Duration | Task Tries | Landing Times | Gantt | Details | Code | Refresh | Delete

running Base date: 2019-10-22 12:06:01 Number of runs: 5 Run: scheduled__2019-10-22T12:06:00+00:00 Layout: Left->Right Go

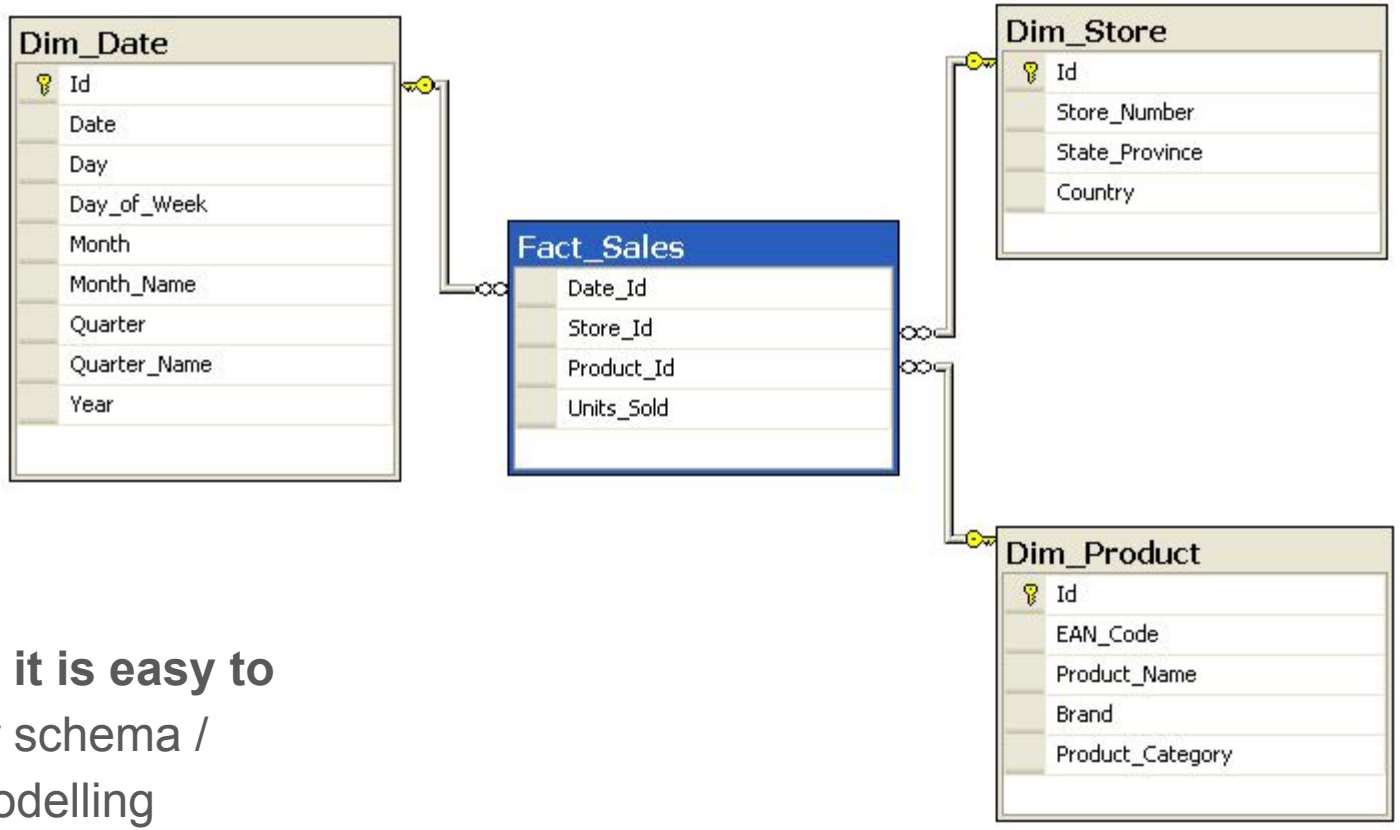
PythonProgramOperator



From Aakash Pydi's Medium Post : Building a Production-Level ETL Pipeline Platform Using Apache Airflow

Data pipeline “tries” to create

1. **One environment:** collect data from various sources into one place
2. **Easier to use data**: pre-joined, transformed, using patterns (star schema / dimensional model, snowflake, data vault, ...)
3. **Single-source-of-truth**: Single table as reference for specific definition



Model data so it is easy to analyze: a star schema / dimensional modelling

Data pipeline “tries” to create

1. **One environment**: collect data from various sources into one place
2. **Easier to use data**: pre-joined, transformed, using patterns (star schema / dimensional model, snowflake, data vault, ...)
3. **Single-source-of-truth**: Single table as reference for specific definition
4. **Cleansed data**: exclude testing data, handle different format, deduplicate

Cleanse data so it is easy to use.

Example: cleanse dirty character, make data uniform

```
SELECT
    REGEXP_REPLACE(phone,'[^0-9]','') AS phone,
    LOWER(email) AS email,
FROM table
```

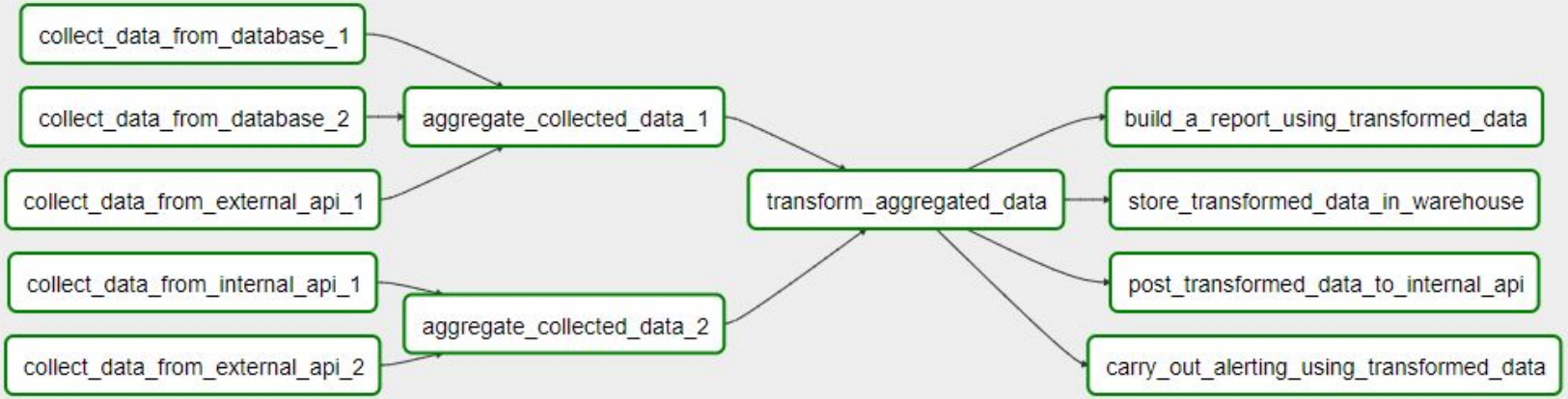
Airflow DAGs Data Profiling Browse Admin Docs About

On DAG: example_dag

Graph View | Tree View | Task Duration | Task Tries | Landing Times | Gantt | Details | Code | Refresh | Delete

running Base date: 2019-10-22 12:06:01 Number of runs: 5 Run: scheduled__2019-10-22T12:06:00+00:00 Layout: Left->Right Go

PythonProgramOperator



From Aakash Pydi's Medium Post : Building a Production-Level ETL Pipeline Platform Using Apache Airflow

```python
from airflow import DAG
from airflow.contrib.operators.bigquery_operator import BigQueryOperator

default_args = {
    "start_date": datetime(2020, 1, 1),
    "schedule_interval": "@daily",
}

with DAG(dag_id="user_data_users", default_args=default_dag_args) as dag:
    ...
    sql = """
      SELECT
        LOWER(email) AS email
      FROM `user_data.users`
    """

    transform_data = BigQueryOperator(
        task_id="transform_data",
        sql=sql,
        destination_dataset_table="cleansed_data.users",
        ...
        dag=dag,
    )
    ...
    extract_data >> transform_data >> another_transform_data
```

Data pipeline “tries” to create

1. **One environment**: collect data from various sources into one place
2. **Easier to use data**: pre-joined, transformed, using patterns (star schema / dimensional model, snowflake, data vault, ...)
3. **Single-source-of-truth**: Single table as reference for specific definition
4. **Cleansed data**: exclude testing data, handle different format, deduplicate
5. **Catalog/Dictionary**: clearly described table along with column descriptions
6. **Cost efficiency**: eliminates the need to process in each of the team

Google Cloud Platform | My Project | Search products and

Data Catalog | Search | CREATE

covid | SEARCH

Sort by: Relevance | Systems | Data types | Include public datasets

Name	Description	Type	System	Project	Last modified
summary Dataset: covid19_jhu_csse	Summary COVID-19 cases, aggregated by country/region and province/state. See the original source files here: ...	Table	BigQuery	bigquery-public-data	Jul 22, 2020
mobility_report Dataset: covid19_google_mobility	This dataset is intended to help remediate the impact of COVID-19. It shouldn't be used for medical diagnostic, prognostic, or treatment purposes. It also isn't ...	Table	BigQuery	bigquery-public-data	Jul 22, 2020
covid19_open_data Dataset: covid19_open_data	This dataset contains country-level datasets of daily time-series data related to COVID-19 globally. You can	Table	BigQuery	bigquery-public-data	Jul 22, 2020

Data pipeline “tries” to create

1. **One environment**: collect data from various sources into one place
2. **Easier to use data**: pre-joined, transformed, using patterns (star schema / dimensional model, snowflake, data vault, ...)
3. **Single-source-of-truth**: Single table as reference for specific definition
4. **Cleansed data**: exclude testing data, handle different format, deduplicate
5. **Catalog/Dictionary**: clearly described table along with column descriptions
6. **Cost efficiency**: eliminates the need to process in each of the team

But sometimes it does not solve the problems...

- **Yet another copy of data**, but old data still being used
- Not being used because **hard to understand**
- Not being used because **not trusted** (different number)
- **Failing often**, not reliable, not timely
- **Takes long time to implement**, users end up creating their own data pipeline

Hard to use. Not reliable. Hard to maintain.
Not so enjoyable...

Principles of Enjoyable Data Pipeline

Enjoyable Data Pipeline

1. **Ease of Use** - Gampang dipakai data analysts / data scientists
2. **Trustworthy** - Data yang disajikan berkualitas, bisa dipercaya
3. **Maintainable** - Kode mudah dibaca, mudah diubah, dan didebug

Ease of Use

Ease of Use

- **Close collaboration** with user: Does this make sense to you?
- Tables are **intuitive**. Easy to query: Try to create one dashboard and feel it
- Nice **design trade off** between ease of query and modeling best practice
- Clarity of **definition**
- **Searchable**
- **Verbose** name: descriptive, clear timezone

Trustworthy

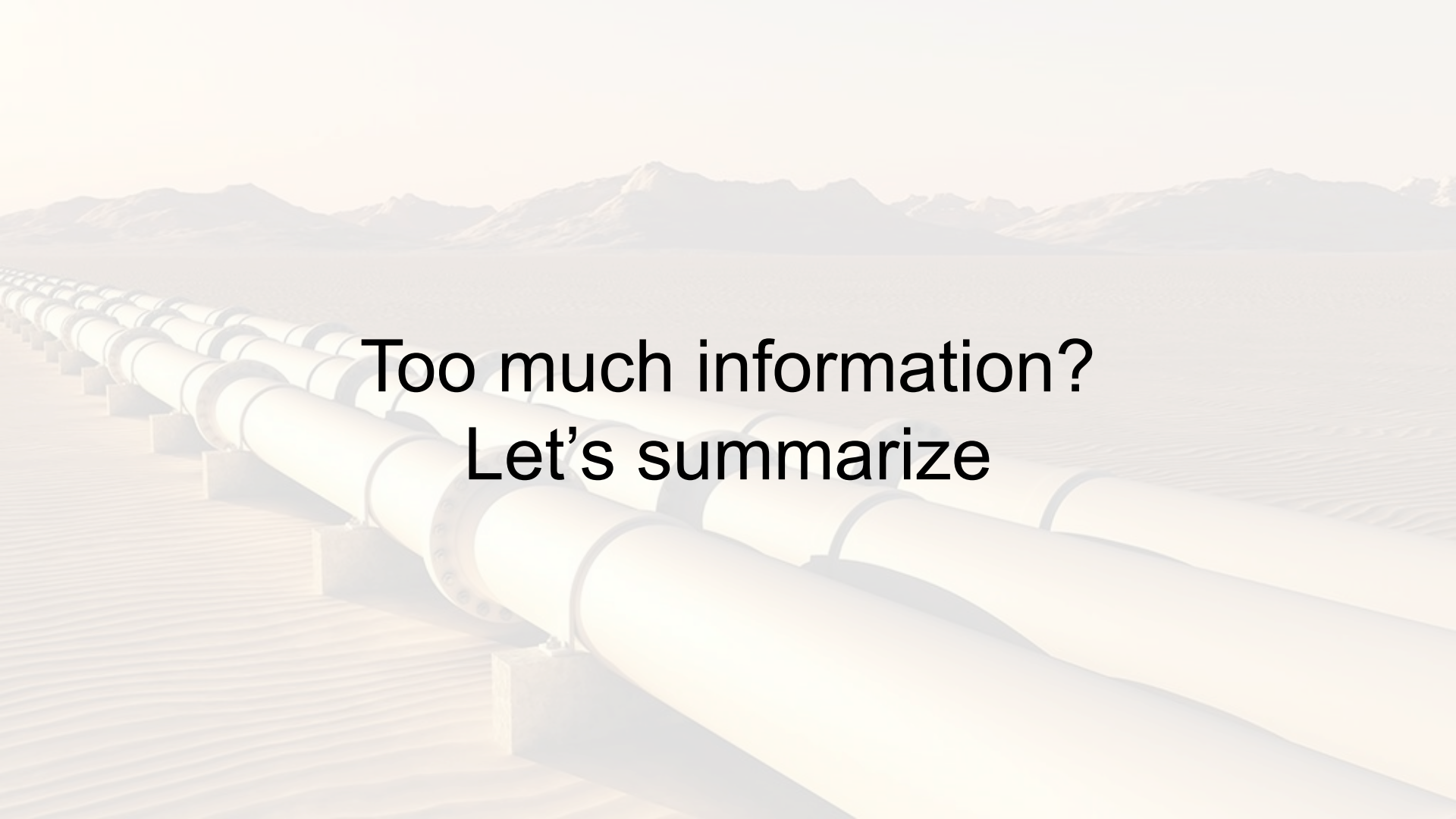
Trustworthy

- Again, **close collaboration** with users: testings, sharing numbers
- **Human-centric** Operations: Over-communicate, active update when issue happens, manage expectation
- Fulfill **Quality**: Timeliness, completeness, uniqueness, consistency
- **Code** Review, Testing
- **SLA**, Alerting, On-Call

Maintainable

Maintainable

- Good **Data Layers** Design: staging → … → warehouse
- **Readable** codes / SQL: alias & CTE naming, indentation, capital
- Clear data **modeling convention**: fact, dim
- **Divide** and Conquer: avoid >300 lines of SQLs
- **Automated Testing**: uniqueness, compleness, validation
- **Idempotent** Processing: deterministic when we rerun



Too much information?
Let’s summarize

Recap

Principles of Enjoyable Data Pipeline

1. **Ease of Use**
2. **Trustworthy**
3. **Maintainable**

Most important things to remember:

1. **Collaborate** closely with user
2. **Human-centric** operations
3. **Craftmanship** on building data pipeline

That’s all folks.

@rendybjunior | hi@rendyistyping.com