## Overview

Untuk Final Project, kita akan membuat end-to-end data pipeline yang mengambil data dari berbagai source dan mengolahnya sehingga menjadi data yang clean dan usable.

Mempertimbangkan waktu pengerjaan dan integrasi dengan data pipeline, berikut materi yang akan di include pada Final Project:

- Python

- Github

- SQL

- NoSQL

- Docker

- Spark

- Airflow

- Kafka

- ML

Materi yang di exclude:

- API

- Java

- MapReduce

- Hadoop

- BI

## Prerequisite

Ada beberapa tools yang perlu di install (docker) untuk Final Project kali ini. Tools yang perlu di install adalah:

- Python (Local) [Link](https://conda.io/projects/conda/en/latest/user-guide/install/index.html)

- Docker (Local) [Link](https://www.docker.com/)

- Git (Local) [Link](https://git-scm.com/)

- Airflow (Docker) [Link](https://airflow.apache.org/docs/apache-airflow/stable/howto/docker-compose/index.html)

- MySQL (Docker) [Link](https://hub.docker.com/_/mysql)

- PostgreSQL (Docker) [Link](https://hub.docker.com/_/postgres)

- Spark (Docker) [Link](https://hub.docker.com/r/bitnami/spark/)

- Kafka (Docker) [Link](https://github.com/wurstmeister/kafka-docker)

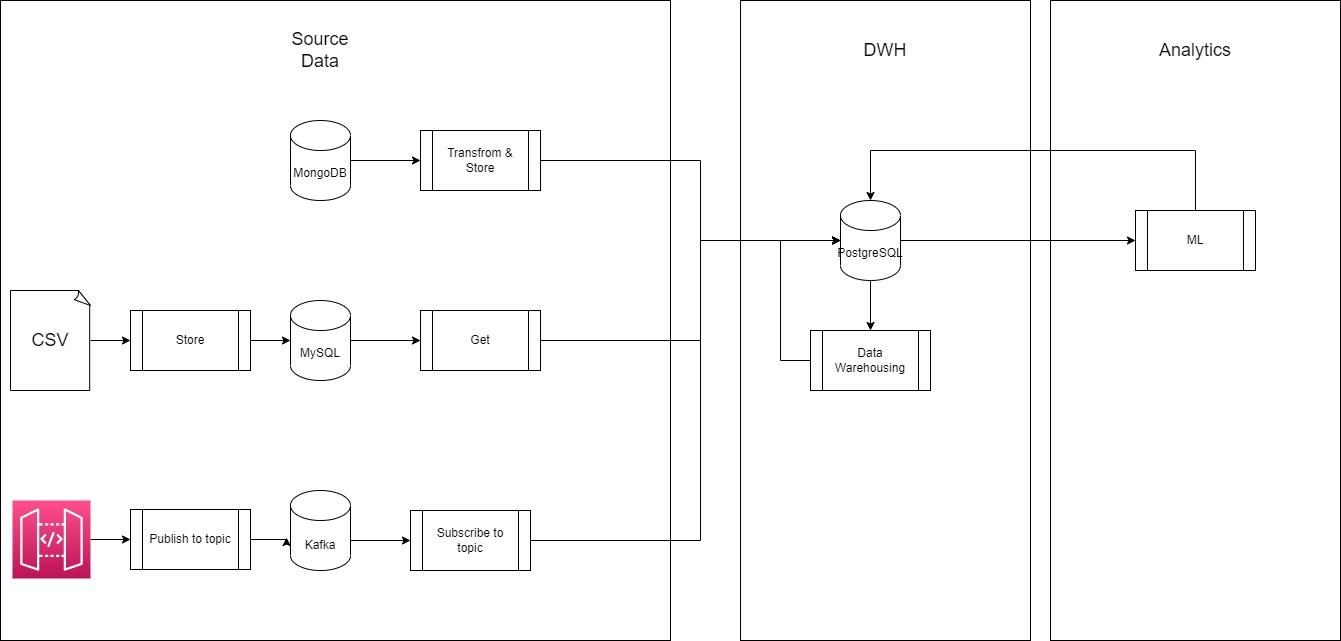
Dan ada pula tools yang bisa dibuat melalui web, yaitu:

- MongoDB (MongoDB Atlas)

## Project

### General Flow

1. Pipeline batch (using python & spark)
2. Pipeline real time (using kafka)
3. Data warehousing (SQL)
4. Machine Learning



1.1 Get data from MySQL

populate MySQL from CSV ML data (Spark)

Table name:

* home\_credit\_default\_risk\_application\_test [Link](https://www.kaggle.com/competitions/home-credit-default-risk/data?select=application_test.csv)
* home\_credit\_default\_risk\_application\_train [Link](https://www.kaggle.com/competitions/home-credit-default-risk/data?select=application_train.csv)

Get MySQL data to PostgreSQL (Spark). Same table name.

1.2 Get data from MongoDB

Populate MongoDB (using sample data)

Transform sample\_training.zips (flatten lat long)

Transform sample\_training.companies (exclude nested data, only first **offices**)

Table name:

* zips
* companies

Get transformed data to PostgreSQL (Python)

2.1 Stream kafka topic

Create kafka topic = TopicCurrency

Create script to push realtime (minutely) public API data to topic (<https://freeforexapi.com/Home/Api>)

API: <https://www.freeforexapi.com/api/live?pairs=EURUSD,EURGBP,USDEUR>

script to subscribe topic and send data to postgresql (add details currency)

EURUSD=US Dollar, EURGBP=Pound Sterling, USDEUR=Euro

Format = {‘currency\_id’, ‘rate’, ‘timestamp’, ‘currency\_name’}

3.1 Create dim table

Create dim country (id, country\_code)

Create dim state (id, country\_id, state\_code)

Create dim city (id, state\_id, city\_name, zip\_code)

Create dim currency (id, currency\_name, currency\_code)

3.2 Create fact table (daily and monthly airflow schedule)

Create fact total city & office per state

Create fact currency daily avg (using macros previous day)

Create fact currency monthly avg (using macros previous month)

4.1 Preprocessing

get data from DB

Check train data label distribution

Check the null column. If > 60% null, drop column.

one hot encoding categorical data

impute null values with median

Store data to table:

* home\_credit\_default\_risk\_application\_test\_clean
* home\_credit\_default\_risk\_application\_train\_clean

4.2 Logistic Regression

fit ML

store ID & probability result to DB

Table name: home\_credit\_default\_risk\_application\_ml\_result

### Airflow Dependency

DAG Unscheduled (schedule=None)

DAG MongoDB

transform\_mongo

DAG MySQL

get\_csv >> get\_mysql

DAG dim (dependency DAG MongoDB & DAG MySQL)

1. create\_dim\_table

2. create\_dim\_table\_1 >> create\_dim\_table\_2

DAG Daily

DAG fact\_daily (dependency DAG dim)

create\_fact\_total\_state >> create\_fact\_currency\_daily

DAG Monthly

DAG fact\_monthly (dependency DAG dim)

create\_fact\_currency\_monthly

### Penilaian

1. Pemahaman flow end-to-end data pipeline
2. Kesesuaian dengan project requirement
3. Kerapian code