Orchestrating ETL for School Data Using Airflow

From API Data Extraction to PostgreSQL Database Integration

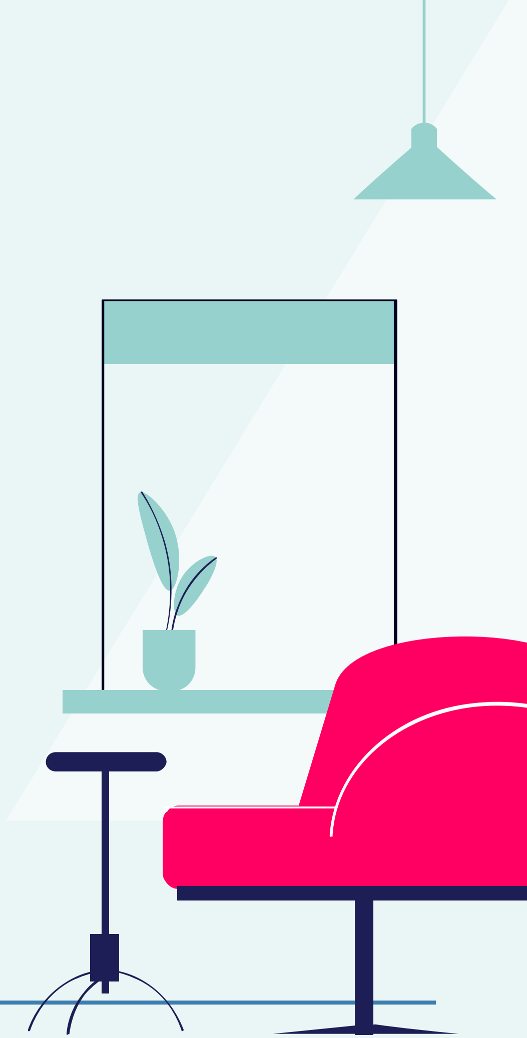
Hijir Della Wirasti
Business Intelligence
Batch 13
21 December 2024

Resource

Github	https://github.com/hijirdella/ETL-Pipeline-Airflow-School-API
API	https://api-sekolah-indonesia.vercel.app/sekolah?page=1&perPage=2000
LinkedIn	https://www.linkedin.com/in/hijirdella/
Email	hijirdw@gmail.com

Thanks to My Mentor: **Mohamad Ikhsan Nurulloh**

Table of contents

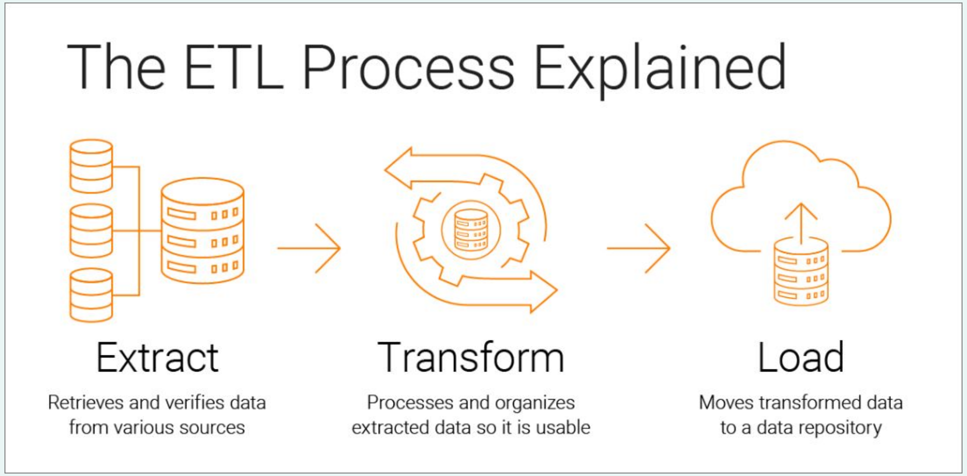


01 Introduction to ETL with Apache Airflow

What is ETL?

Extract, Transform, Load (ETL) is a data integration process involving:

- **Extract**: Collecting data from various sources such as APIs, databases, or files.
- **Transform**: Cleaning, filtering, and organizing the data to meet specific requirements.
- **Load**: Inserting the prepared data into a target system, such as a database.


The ETL Process Explained
Extract
Retrieves and verifies data from various sources
Transform
Processes and organizes extracted data so it is usable
Load
Moves transformed data to a data repository

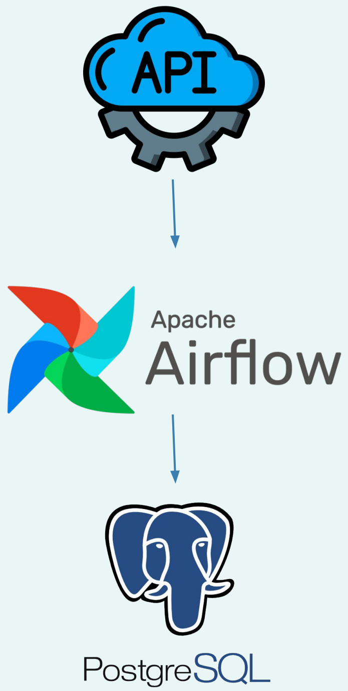
Why Apache Airflow?

- **Overview of Apache Airflow**:
  - An open-source workflow orchestration tool designed for automating complex workflows like ETL.
  - Allows users to define workflows as Directed Acyclic Graphs (DAGs).
- **Key Features of Apache Airflow**:
  - **Modular Workflows**: Tasks are defined as Python code for easy customization.
  - **Scheduling**: Automates workflows to run at specific intervals.
  - **Monitoring**: Provides a web-based interface for tracking task progress, logs, and statuses.
  - **Scalability**: Supports distributed execution for large datasets or complex pipelines.

02 ETL from API to Database

Objective

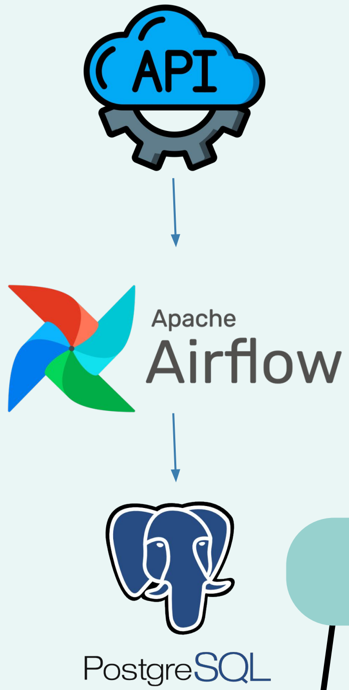
- **Goal**: Automate the ETL process to seamlessly extract, transform, and load school data into a PostgreSQL database.
- **Key Outcomes**:
  - Retrieve school data from a public API.
  - Apply necessary transformations for cleaning and restructuring the data.
  - Store the processed data into a PostgreSQL database for analysis and reporting.



03 Task Overview

Tasks

1. **Fetching Data from APIs**:
   - Used the School Data API (`https://api-sekolah-indonesia.vercel.app/sekolah`) to retrieve raw JSON data.
   - The API request fetches up to 2,000 records per call, providing detailed school data such as names, locations, and types.
2. **Transformation of Retrieved Data**:
   - Filtered data for **SMA (high schools)** with a **status of "N" (National/State-owned)**.
   - Added a new column `school_address` combining school name and address for better usability.
   - Converted latitude (`lintang`) and longitude (`bujur`) values to numeric formats for validation.
   - Removed rows with invalid or missing latitude and longitude values.
3. **Loading Transformed Data into PostgreSQL Database**:
   - Created a PostgreSQL table named `target_table` with appropriate schema.
   - Inserted transformed data into the database, replacing existing data during each DAG run to ensure consistency.
   - Verified that the table schema matches the data types of the transformed data.



04 Implementation - Build DAG (1)

Task 1: Fetch Data from API

- **Function**: `fetch_data_from_api`
  - Uses the `requests` library to send an HTTP GET request to the School API endpoint: `https://api-sekolah-indonesia.vercel.app/sekolah?page=1&perPage=5000`.
  - Retrieves data in JSON format, including school names, types, locations, and statuses.
  - Passes the fetched data to the next task via Airflow's XCom.

```python
from datetime import datetime, timedelta
from airflow import DAG
from airflow.operators.python_operator import PythonOperator
from airflow.providers.postgres.hooks.postgres import PostgresHook
import pandas as pd
import requests

# Define default arguments
default_args = {
    'owner': 'hijir',
    'depends_on_past': False,
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
    'start_date': datetime(2024, 5, 1)
}

# Function to fetch data from API
def fetch_data_from_api():
    response =
requests.get('https://api-sekolah-indonesia.vercel.app/sekolah?page=1&perPage=5
000')
    data = response.json()
    return data
```

04 Implementation - Build DAG (2)

Task 2: Transform Data

- **Function**: `transform_data`
  - Reads the raw JSON data fetched from the API and converts it into a Pandas DataFrame.
  - **Transformations Applied**:
    - Filters for **SMA (high schools)** with a **status of "N" (National/State-owned)**.
    - Adds a new column `school_address` by combining the school name and address for better readability.
    - Converts latitude (`lintang`) and longitude (`bujur`) values to numeric types for consistency.
    - Removes rows where latitude or longitude values are missing.
  - Outputs the cleaned and transformed DataFrame to the next task.
  -

```python
# Function to transform data
def transform_data (**kwargs):
    data = kwargs['ti'].xcom_pull( task_ids='fetch_data_from_api' )
    # Extract the relevant data from the nested structure
    sekolah_data  = data['dataSekolah' ]
    # Transform JSON data to DataFrame
    df = pd.DataFrame (sekolah_data )

    # Transformation 1
    # Filter and transform data using pandas
    transformed_df  = df[(df['status' ].str.contains ('N')) & (df['bentuk' ] == 'SMA')]

    # Transformation 2
    # Tambahkan kolom yang menggabungkan nama sekolah dan alamat jalan
    transformed_df ['school_address' ] = transformed_df ['sekolah' ] + " - " +
transformed_df ['alamat_jalan' ]

    # Transformation 3
    # Konversi kolom lintang dan bujur menjadi tipe data numerik
    transformed_df ['lintang' ] = pd.to_numeric (transformed_df ['lintang' ],
errors='coerce' )
    transformed_df ['bujur' ] = pd.to_numeric (transformed_df ['bujur' ], errors='coerce' )

    # Transformation 4
    # Hapus baris dengan data lintang atau bujur yang kosong
    transformed_df  = transformed_df .dropna (subset=['lintang' , 'bujur' ])

    return  transformed_df
```

04 Implementation - Build DAG (3)

Task 3: Load Data to Database

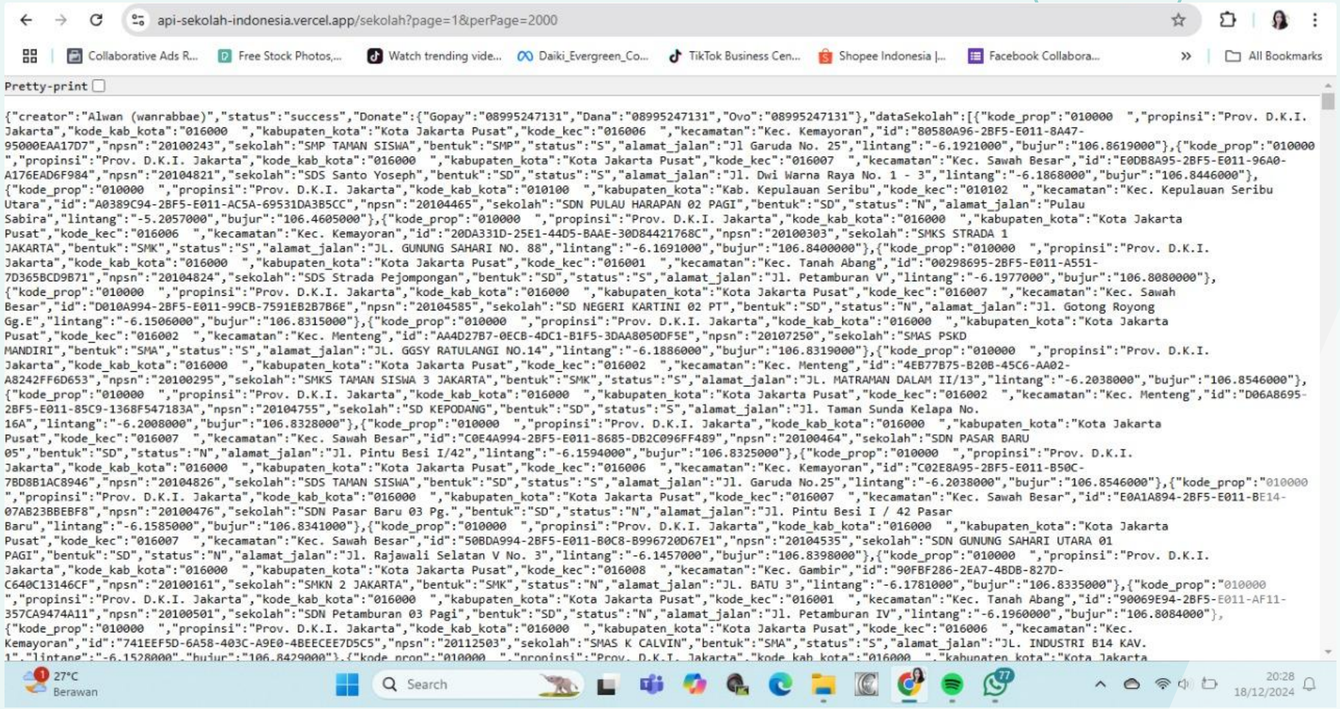
- **Function**: `load_data_to_database`
  - Uses Airflow's `PostgresHook` to connect to the PostgreSQL database.
  - Automatically creates a table (`target_table`) in the specified schema (`hijir`) if it does not exist.
  - Loads the transformed data into the table, replacing old records with each DAG run to maintain consistency.

```python
## Define Schema before load ( Change to your schema name)
custom_schema = 'hijir'

# Function to load data into database
# Define the custom schema name
def load_data_to_database (**kwargs):
    transformed_data = kwargs['ti'].xcom_pull(task_ids='transform_data' )

    # Retrieve connection from Airflow
    postgres_hook = PostgresHook(postgres_conn_id='postgres')

    # Analyze data types of transformed data
    data_types = {col: 'TEXT' for col, dtype in transformed_data.dtypes.items()}

    # Create table if it doesn't exist
    create_query = f"""
    CREATE TABLE IF NOT EXISTS {custom_schema}.target_table (
        {', '.join([f'{col} {data_types[col]}' for col in transformed_data.columns])}
    );
    """
    postgres_hook.run(create_query)

    # Load data into the table with custom schema
    transformed_data.to_sql('target_table', postgres_hook.get_sqlalchemy_engine(),
schema=custom_schema, if_exists='replace', index=False)
```

04 Implementation - Build DAG (4)

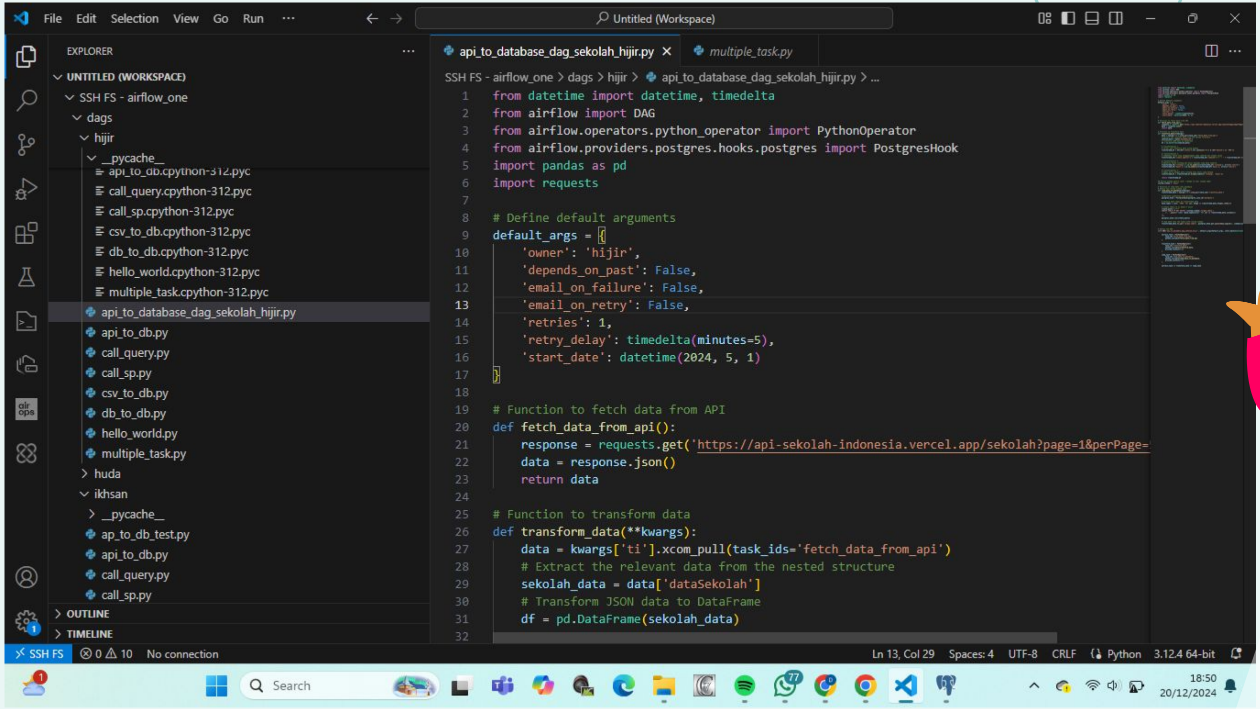
```
# Define the DAG
with DAG('api_to_database_dag_sekolah_hijir' , default_args =default_args ,
start_date =datetime (2024, 5, 1), schedule_interval ='@daily' , catchup=False) as dag:

    extract_task  = PythonOperator(
        task_id='fetch_data_from_api' ,
        python_callable =fetch_data_from_api
    )

    transform_task  = PythonOperator(
        task_id='transform_data' ,
        python_callable =transform_data ,
        provide_context =True
    )

    load_task  = PythonOperator(
        task_id='load_data_to_database' ,
        python_callable =load_data_to_database ,
        provide_context =True
    )

    extract_task  >> transform_task  >> load_task
```

05 Documentation - API



```
{"creator":"Alwan (wanrabbae)","status":"success","Donate":{"Gopay":"08995247131","Dana":"08995247131","Ovo":"08995247131"},"dataSekolah":[{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. 
Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016006  ","kecamatan":"Kec. Kemayoran","id":"80580A96-2BF5-E011-8A47-
95000EAA17D7","npsn":"20100243","sekolah":"SMP TAMAN SISWA","bentuk":"SMP","status":"S","alamat_jalan":"Jl Garuda No. 25","lintang":"-6.1921000","bujur":"106.8619000"},{"kode_prop":"010000
","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016007  ","kecamatan":"Kec. Sawah Besar","id":"E0DB8A95-2BF5-E011-96A0-
A176EAD6F984","npsn":"20104821","sekolah":"SDS Santo Yoseph","bentuk":"SD","status":"S","alamat_jalan":"Jl. Dwi Warna Raya No. 1 - 3","lintang":"-6.1868000","bujur":"106.8446000"},
{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"010100  ","kabupaten_kota":"Kab. Kepulauan Seribu","kode_kec":"010102  ","kecamatan":"Kec. Kepulauan Seribu 
Utara","id":"A0389C94-2BF5-E011-AC5A-69531DA3B5CC","npsn":"20104465","sekolah":"SDN PULAU HARAPAN 02 PAGI","bentuk":"SD","status":"N","alamat_jalan":"Pulau 
Sabira","lintang":"-5.2057000","bujur":"106.4605000"},{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta 
Pusat","kode_kec":"016006  ","kecamatan":"Kec. Kemayoran","id":"20DA331D-25E1-44D5-BAAE-30D84421768C","npsn":"20100303","sekolah":"SMKS STRADA 1 
JAKARTA","bentuk":"SMK","status":"S","alamat_jalan":"JL. GUNUNG SAHARI NO. 88","lintang":"-6.1691000","bujur":"106.8400000"},{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. 
Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016001  ","kecamatan":"Kec. Tanah Abang","id":"00298695-2BF5-E011-A551-
7D365BCD9B71","npsn":"20104824","sekolah":"SDS Strada Pejompongan","bentuk":"SD","status":"S","alamat_jalan":"Jl. Petamburan V","lintang":"-6.1977000","bujur":"106.8080000"},
{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016007  ","kecamatan":"Kec. Sawah 
Besar","id":"D010A994-2BF5-E011-99CB-7591EB2B7B6E","npsn":"20104585","sekolah":"SD NEGERI KARTINI 02 PT","bentuk":"SD","status":"N","alamat_jalan":"Jl. Gotong Royong 
Gg.E","lintang":"-6.1506000","bujur":"106.8315000"},{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta 
Pusat","kode_kec":"016002  ","kecamatan":"Kec. Menteng","id":"AA4D27B7-0ECB-4DC1-B1F5-3DAA8050DF5E","npsn":"20107250","sekolah":"SMAS PSKD 
MANDIRI","bentuk":"SMA","status":"S","alamat_jalan":"JL. GGSY RATULANGI NO.14","lintang":"-6.1886000","bujur":"106.8319000"},{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. 
Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016002  ","kecamatan":"Kec. Menteng","id":"4EB77B75-B20B-45C6-AA02-
A8242FF6D653","npsn":"20100295","sekolah":"SMKS TAMAN SISWA 3 JAKARTA","bentuk":"SMK","status":"S","alamat_jalan":"JL. MATRAMAN DALAM II/13","lintang":"-6.2038000","bujur":"106.8546000"},
{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016002  ","kecamatan":"Kec. Menteng","id":"D06A8695-
2BF5-E011-85C9-1368F547183A","npsn":"20104755","sekolah":"SD KEPODANG","bentuk":"SD","status":"S","alamat_jalan":"Jl. Taman Sunda Kelapa No. 
16A","lintang":"-6.2008000","bujur":"106.8328000"},{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta 
Pusat","kode_kec":"016007  ","kecamatan":"Kec. Sawah Besar","id":"C0E4A994-2BF5-E011-8685-DB2C096FF489","npsn":"20100464","sekolah":"SDN PASAR BARU 
05","bentuk":"SD","status":"N","alamat_jalan":"Jl. Pintu Besi I/42","lintang":"-6.1594000","bujur":"106.8325000"},{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. 
Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016006  ","kecamatan":"Kec. Kemayoran","id":"C02E8A95-2BF5-E011-B50C-
7BD8B1AC8946","npsn":"20104826","sekolah":"SDS TAMAN SISWA","bentuk":"SD","status":"S","alamat_jalan":"Jl. Garuda No.25","lintang":"-6.2038000","bujur":"106.8546000"},{"kode_prop":"010000
","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016007  ","kecamatan":"Kec. Sawah Besar","id":"E0A1A894-2BF5-E011-BE14-
07AB23BBEBF8","npsn":"20100476","sekolah":"SDN Pasar Baru 03 Pg.","bentuk":"SD","status":"N","alamat_jalan":"Jl. Pintu Besi I / 42 Pasar 
Baru","lintang":"-6.1585000","bujur":"106.8341000"},{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta 
Pusat","kode_kec":"016007  ","kecamatan":"Kec. Sawah Besar","id":"50BDA994-2BF5-E011-B0C8-B996720D67E1","npsn":"20104535","sekolah":"SDN GUNUNG SAHARI UTARA 01 
PAGI","bentuk":"SD","status":"N","alamat_jalan":"Jl. Rajawali Selatan V No. 3","lintang":"-6.1457000","bujur":"106.8398000"},{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. 
Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016008  ","kecamatan":"Kec. Gambir","id":"90FBF286-2EA7-4BDB-827D-
C640C13146CF","npsn":"20100161","sekolah":"SMKN 2 JAKARTA","bentuk":"SMK","status":"N","alamat_jalan":"JL. BATU 3","lintang":"-6.1781000","bujur":"106.8335000"},{"kode_prop":"010000
","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016001  ","kecamatan":"Kec. Tanah Abang","id":"90069E94-2BF5-E011-AF11-
357CA9474A11","npsn":"20100501","sekolah":"SDN Petamburan 03 Pagi","bentuk":"SD","status":"N","alamat_jalan":"Jl. Petamburan IV","lintang":"-6.1960000","bujur":"106.8084000"},
{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta Pusat","kode_kec":"016006  ","kecamatan":"Kec. 
Kemayoran","id":"741EEF5D-6A58-403C-A9E0-4BEECEE7D5C5","npsn":"20112503","sekolah":"SMAS K CALVIN","bentuk":"SMA","status":"S","alamat_jalan":"JL. INDUSTRI B14 KAV. 
1","lintang":"-6.1528000","bujur":"106.8429000"},{"kode_prop":"010000  ","propinsi":"Prov. D.K.I. Jakarta","kode_kab_kota":"016000  ","kabupaten_kota":"Kota Jakarta
```

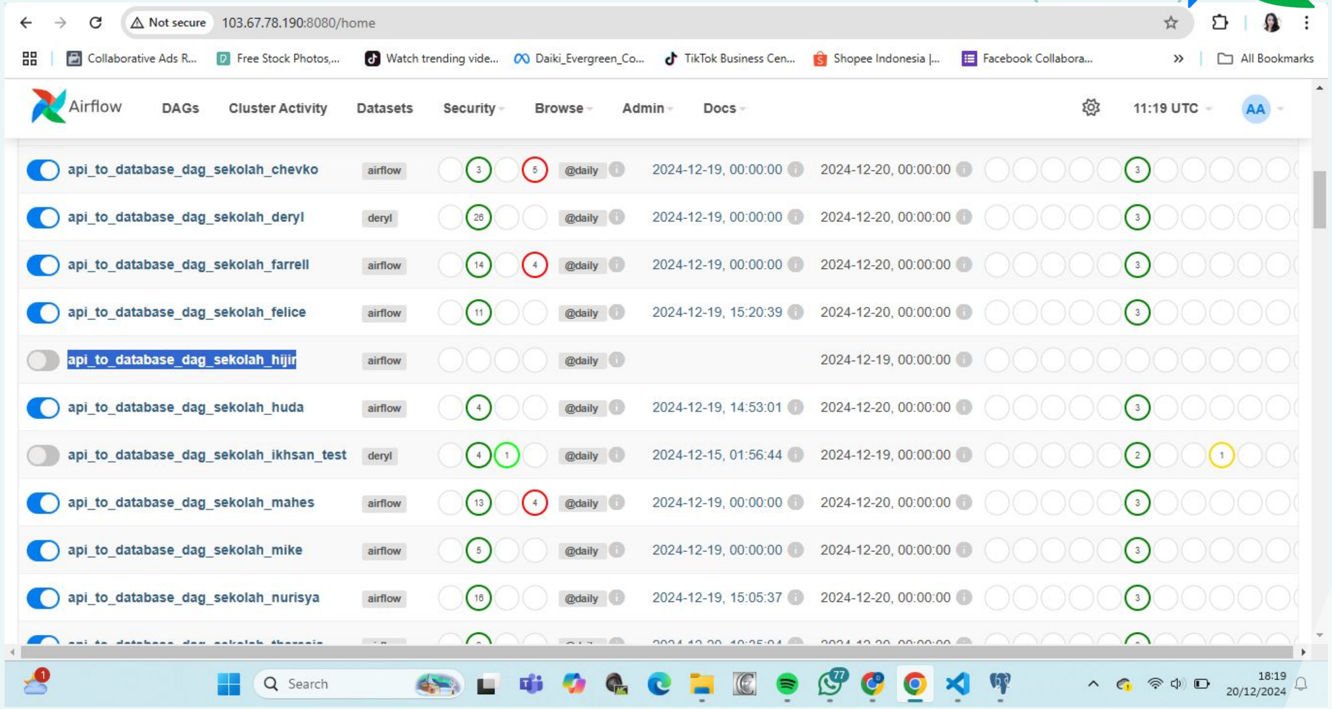
05 Documentation - Build DAG


File Edit Selection View Go Run ···
Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
SSH FS - airflow_one
dags
hijir
pycache
api_to_db.cpython-312.pyc
call_query.cpython-312.pyc
call_sp.cpython-312.pyc
csv_to_db.cpython-312.pyc
db_to_db.cpython-312.pyc
hello_world.cpython-312.pyc
multiple_task.cpython-312.pyc
api_to_database_dag_sekolah_hijir.py
api_to_db.py
call_query.py
call_sp.py
csv_to_db.py
db_to_db.py
hello_world.py
multiple_task.py
huda
ikhsan
pycache
ap_to_db_test.py
api_to_db.py
call_query.py
call_sp.py
OUTLINE
TIMELINE
api_to_database_dag_sekolah_hijir.py
multiple_task.py
SSH FS - airflow_one > dags > hijir > api_to_database_dag_sekolah_hijir.py > ...
SSH FS
0 10 No connection
Ln 13, Col 29 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit
Search
18:50
20/12/2024

```python
from datetime import datetime, timedelta
from airflow import DAG
from airflow.operators.python_operator import PythonOperator
from airflow.providers.postgres.hooks.postgres import PostgresHook
import pandas as pd
import requests

# Define default arguments
default_args = {
    'owner': 'hijir',
    'depends_on_past': False,
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
    'start_date': datetime(2024, 5, 1)
}

# Function to fetch data from API
def fetch_data_from_api():
    response = requests.get('https://api-sekolah-indonesia.vercel.app/sekolah?page=1&perPage=
    data = response.json()
    return data

# Function to transform data
def transform_data(**kwargs):
    data = kwargs['ti'].xcom_pull(task_ids='fetch_data_from_api')
    # Extract the relevant data from the nested structure
    sekolah_data = data['dataSekolah']
    # Transform JSON data to DataFrame
    df = pd.DataFrame(sekolah_data)
```

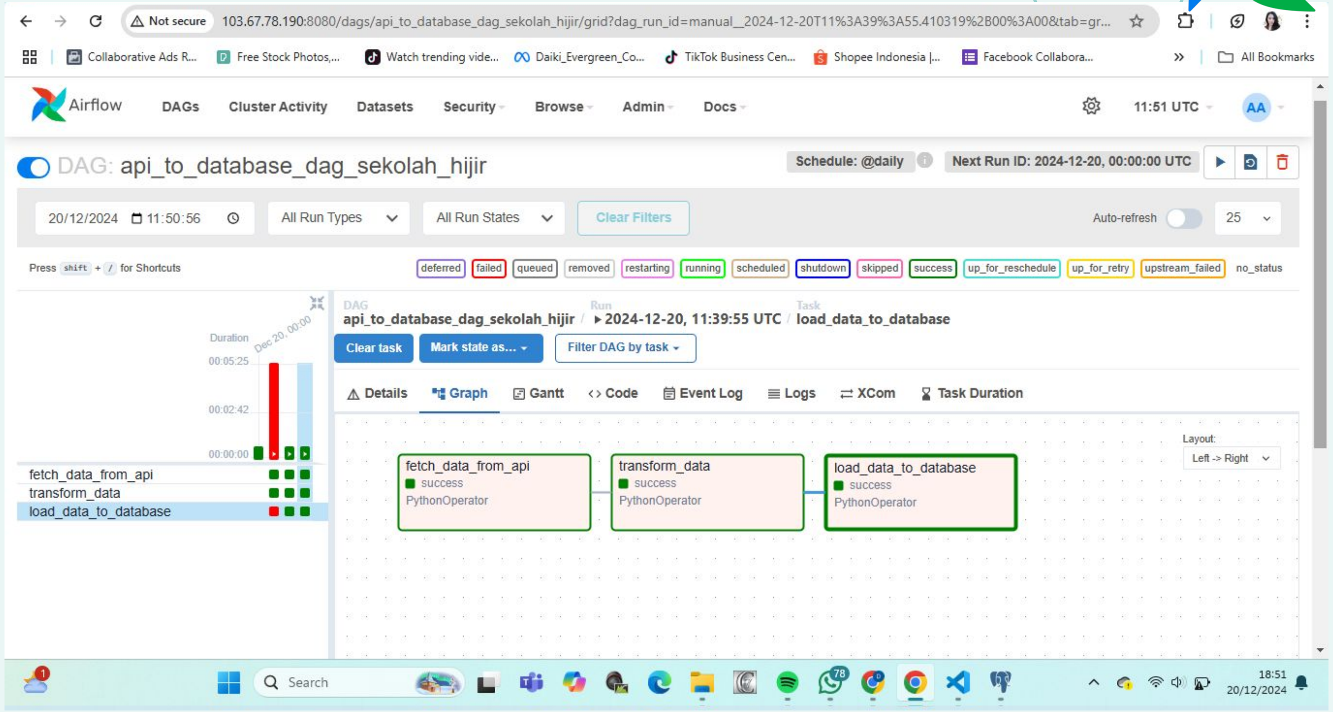
05 Documentation - DAG

Apache Airflow



api_to_database_dag_sekolah_chevko	airflow	3 5	@daily	2024-12-19, 00:00:00	2024-12-20, 00:00:00	3
api_to_database_dag_sekolah_deryl	deryl	26	@daily	2024-12-19, 00:00:00	2024-12-20, 00:00:00	3
api_to_database_dag_sekolah_farrell	airflow	14 4	@daily	2024-12-19, 00:00:00	2024-12-20, 00:00:00	3
api_to_database_dag_sekolah_felice	airflow	11	@daily	2024-12-19, 15:20:39	2024-12-20, 00:00:00	3
api_to_database_dag_sekolah_hijir	airflow		@daily		2024-12-19, 00:00:00	
api_to_database_dag_sekolah_huda	airflow	4	@daily	2024-12-19, 14:53:01	2024-12-20, 00:00:00	3
api_to_database_dag_sekolah_ikhsan_test	deryl	4 1	@daily	2024-12-15, 01:56:44	2024-12-19, 00:00:00	2 1
api_to_database_dag_sekolah_mahes	airflow	13 4	@daily	2024-12-19, 00:00:00	2024-12-20, 00:00:00	3
api_to_database_dag_sekolah_mike	airflow	5	@daily	2024-12-19, 00:00:00	2024-12-20, 00:00:00	3
api_to_database_dag_sekolah_nurisya	airflow	16	@daily	2024-12-19, 15:05:37	2024-12-20, 00:00:00	3

05 Documentation - DAG



DAG: api_to_database_dag_sekolah_hijir

Schedule: @daily

Next Run ID: 2024-12-20, 00:00:00 UTC

20/12/2024 11:50:56 | All Run Types | All Run States | 25

DAG: api_to_database_dag_sekolah_hijir / Run: 2024-12-20, 11:39:55 UTC / Task: load_data_to_database

- fetch_data_from_api
- transform_data
- load_data_to_database

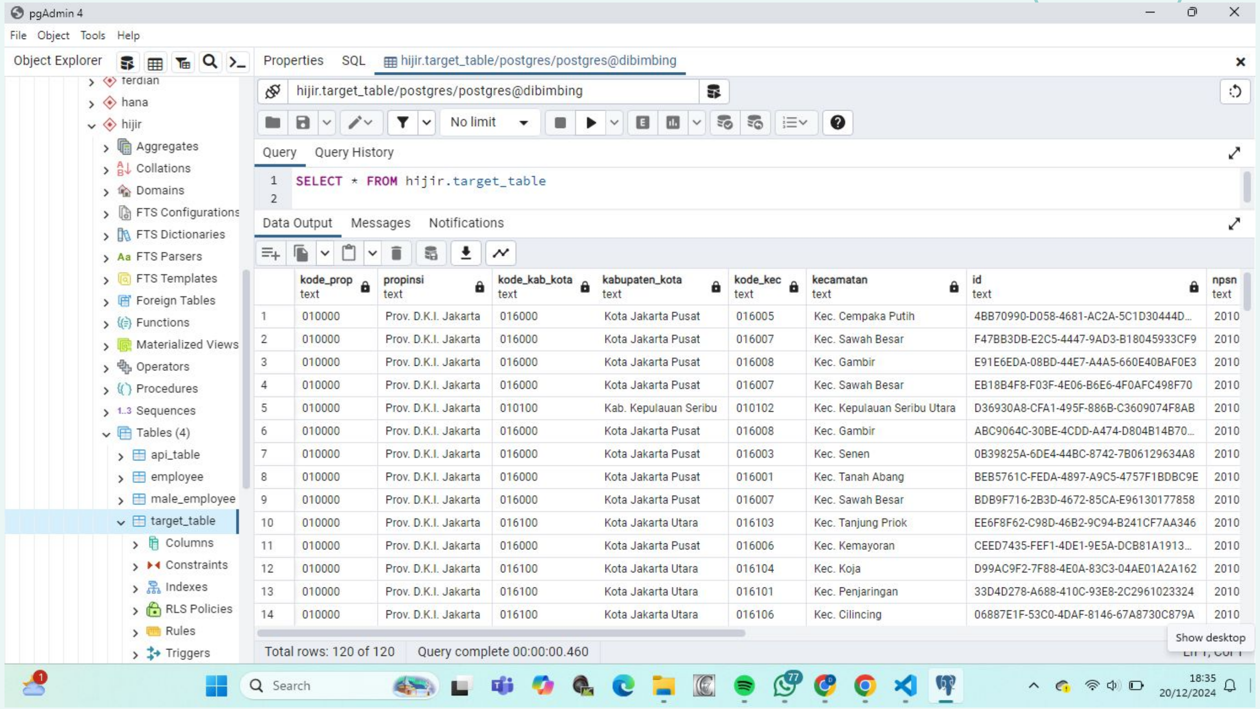
fetch_data_from_api — success — PythonOperator

transform_data — success — PythonOperator

load_data_to_database — success — PythonOperator

Layout: Left -> Right

05 Documentation - Database



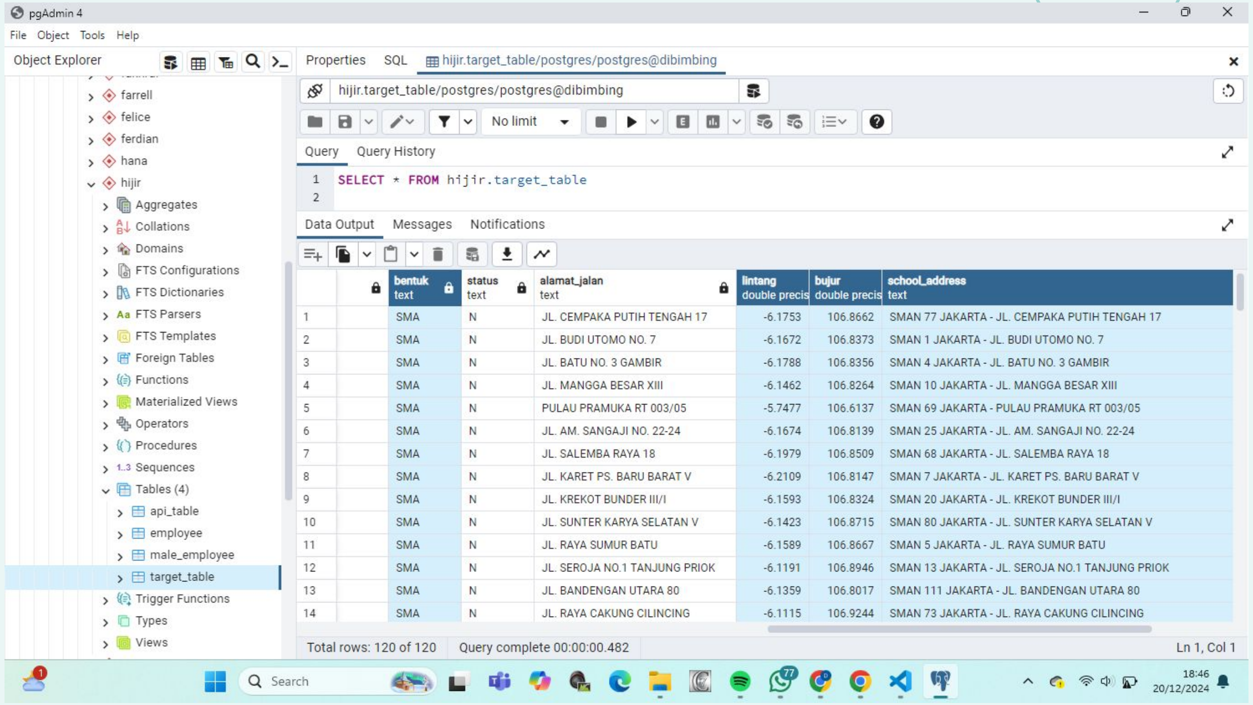
```sql
SELECT * FROM hijir.target_table
```

	kode_prop text	propinsi text	kode_kab_kota text	kabupaten_kota text	kode_kec text	kecamatan text	id text	npsn text
1	010000	Prov. D.K.I. Jakarta	016000	Kota Jakarta Pusat	016005	Kec. Cempaka Putih	4BB70990-D058-4681-AC2A-5C1D30444D...	2010
2	010000	Prov. D.K.I. Jakarta	016000	Kota Jakarta Pusat	016007	Kec. Sawah Besar	F47BB3DB-E2C5-4447-9AD3-B18045933CF9	2010
3	010000	Prov. D.K.I. Jakarta	016000	Kota Jakarta Pusat	016008	Kec. Gambir	E91E6EDA-08BD-44E7-A4A5-660E40BAF0E3	2010
4	010000	Prov. D.K.I. Jakarta	016000	Kota Jakarta Pusat	016007	Kec. Sawah Besar	EB18B4F8-F03F-4E06-B6E6-4F0AFC498F70	2010
5	010000	Prov. D.K.I. Jakarta	010100	Kab. Kepulauan Seribu	010102	Kec. Kepulauan Seribu Utara	D36930A8-CFA1-495F-886B-C3609074F8AB	2010
6	010000	Prov. D.K.I. Jakarta	016000	Kota Jakarta Pusat	016008	Kec. Gambir	ABC9064C-30BE-4CDD-A474-D804B14B70...	2010
7	010000	Prov. D.K.I. Jakarta	016000	Kota Jakarta Pusat	016003	Kec. Senen	0B39825A-6DE4-44BC-8742-7B06129634A8	2010
8	010000	Prov. D.K.I. Jakarta	016000	Kota Jakarta Pusat	016001	Kec. Tanah Abang	BEB5761C-FEDA-4897-A9C5-4757F1BDBC9E	2010
9	010000	Prov. D.K.I. Jakarta	016000	Kota Jakarta Pusat	016007	Kec. Sawah Besar	BDB9F716-2B3D-4672-85CA-E96130177858	2010
10	010000	Prov. D.K.I. Jakarta	016100	Kota Jakarta Utara	016103	Kec. Tanjung Priok	EE6F8F62-C98D-46B2-9C94-B241CF7AA346	2010
11	010000	Prov. D.K.I. Jakarta	016000	Kota Jakarta Pusat	016006	Kec. Kemayoran	CEED7435-FEF1-4DE1-9E5A-DCB81A1913...	2010
12	010000	Prov. D.K.I. Jakarta	016100	Kota Jakarta Utara	016104	Kec. Koja	D99AC9F2-7F88-4E0A-83C3-04AE01A2A162	2010
13	010000	Prov. D.K.I. Jakarta	016100	Kota Jakarta Utara	016101	Kec. Penjaringan	33D4D278-A688-410C-93E8-2C2961023324	2010
14	010000	Prov. D.K.I. Jakarta	016100	Kota Jakarta Utara	016106	Kec. Cilincing	06887E1F-53C0-4DAF-8146-67A8730C879A	2010

Total rows: 120 of 120 Query complete 00:00:00.460

PostgreSQL

05 Documentation - Transformed Data



hijir.target_table/postgres/postgres@dibimbing

```
SELECT * FROM hijir.target_table
```

		bentuk text	status text	alamat_jalan text	lintang double precis	bujur double precis	school_address text
1		SMA	N	JL. CEMPAKA PUTIH TENGAH 17	-6.1753	106.8662	SMAN 77 JAKARTA - JL. CEMPAKA PUTIH TENGAH 17
2		SMA	N	JL. BUDI UTOMO NO. 7	-6.1672	106.8373	SMAN 1 JAKARTA - JL. BUDI UTOMO NO. 7
3		SMA	N	JL. BATU NO. 3 GAMBIR	-6.1788	106.8356	SMAN 4 JAKARTA - JL. BATU NO. 3 GAMBIR
4		SMA	N	JL. MANGGA BESAR XIII	-6.1462	106.8264	SMAN 10 JAKARTA - JL. MANGGA BESAR XIII
5		SMA	N	PULAU PRAMUKA RT 003/05	-5.7477	106.6137	SMAN 69 JAKARTA - PULAU PRAMUKA RT 003/05
6		SMA	N	JL. AM. SANGAJI NO. 22-24	-6.1674	106.8139	SMAN 25 JAKARTA - JL. AM. SANGAJI NO. 22-24
7		SMA	N	JL. SALEMBA RAYA 18	-6.1979	106.8509	SMAN 68 JAKARTA - JL. SALEMBA RAYA 18
8		SMA	N	JL. KARET PS. BARU BARAT V	-6.2109	106.8147	SMAN 7 JAKARTA - JL. KARET PS. BARU BARAT V
9		SMA	N	JL. KREKOT BUNDER III/I	-6.1593	106.8324	SMAN 20 JAKARTA - JL. KREKOT BUNDER III/I
10		SMA	N	JL. SUNTER KARYA SELATAN V	-6.1423	106.8715	SMAN 80 JAKARTA - JL. SUNTER KARYA SELATAN V
11		SMA	N	JL. RAYA SUMUR BATU	-6.1589	106.8667	SMAN 5 JAKARTA - JL. RAYA SUMUR BATU
12		SMA	N	JL. SEROJA NO.1 TANJUNG PRIOK	-6.1191	106.8946	SMAN 13 JAKARTA - JL. SEROJA NO.1 TANJUNG PRIOK
13		SMA	N	JL. BANDENGAN UTARA 80	-6.1359	106.8017	SMAN 111 JAKARTA - JL. BANDENGAN UTARA 80
14		SMA	N	JL. RAYA CAKUNG CILINCING	-6.1115	106.9244	SMAN 73 JAKARTA - JL. RAYA CAKUNG CILINCING

Total rows: 120 of 120 Query complete 00:00:00.482 Ln 1, Col 1

PostgreSQL

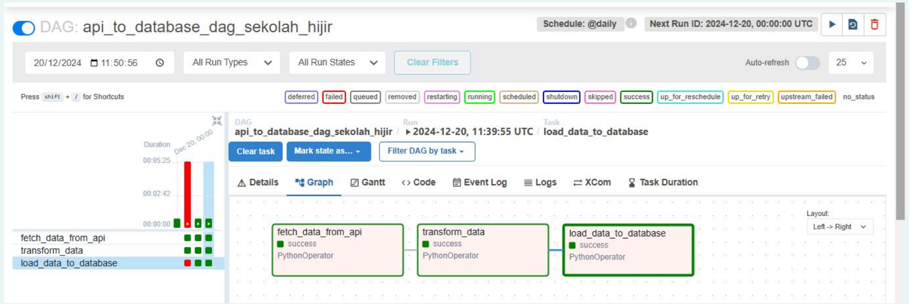
06 Benefits of Using Airflow

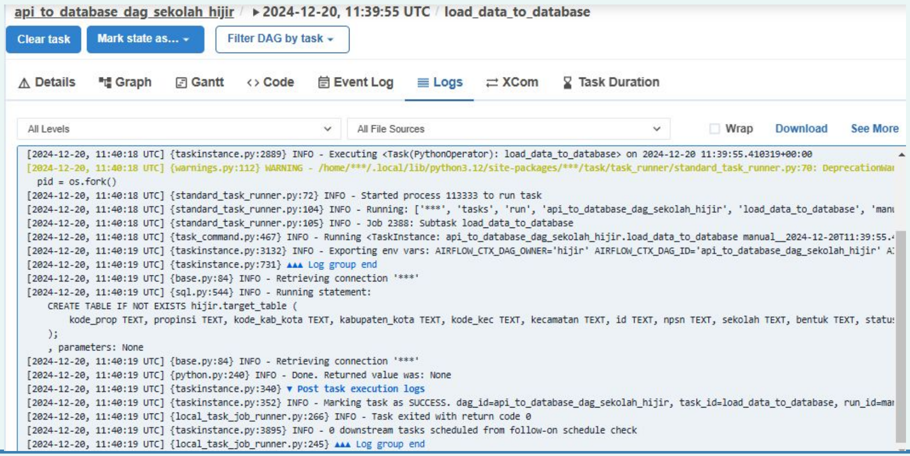
1. **Scalability**:
   - Handles large datasets like school data from an API with thousands of records.
   - Easily integrates additional APIs or data sources into the pipeline.
2. **Scheduling**:
   - Automates daily data updates from the API, ensuring the database always has the latest information.
   - Supports customizable schedules for specific ETL requirements.
3. **Monitoring**:
   - Provides detailed logs for each task, enabling quick debugging of issues.
   - Tracks execution status in real-time via the Airflow web interface, making it easy to identify bottlenecks or errors.
4. **Modularity and Flexibility**:
   - Python-based tasks allow custom transformations, such as filtering for SMA schools and combining columns.
   - DAGs can be updated or extended without disrupting the workflow.
5. **Error Handling and Retries**:
   - Built-in mechanisms for task retries reduce failures due to transient errors (e.g., API timeouts).
   - Error handling ensures data quality by validating and cleaning data during transformations.
6. **Integration**:
   - Seamless connection with PostgreSQL for database operations, automating table creation and data loading.

07 Execution and Testing

Execution of ETL Pipelines

1. **DAG Execution**:
   - The ETL pipeline is executed via Apache Airflow's web interface.
   - Tasks are run sequentially:
     - `fetch_data_from_api` → `transform_data` → `load_data_to_database`.
   - The pipeline fetches, processes, and loads data into PostgreSQL daily, ensuring updated records.
2. **Task Monitoring**:
   - The Airflow web UI shows task status (e.g., running, success, or failed).
   - Logs for each task provide detailed insights into execution and potential errors.



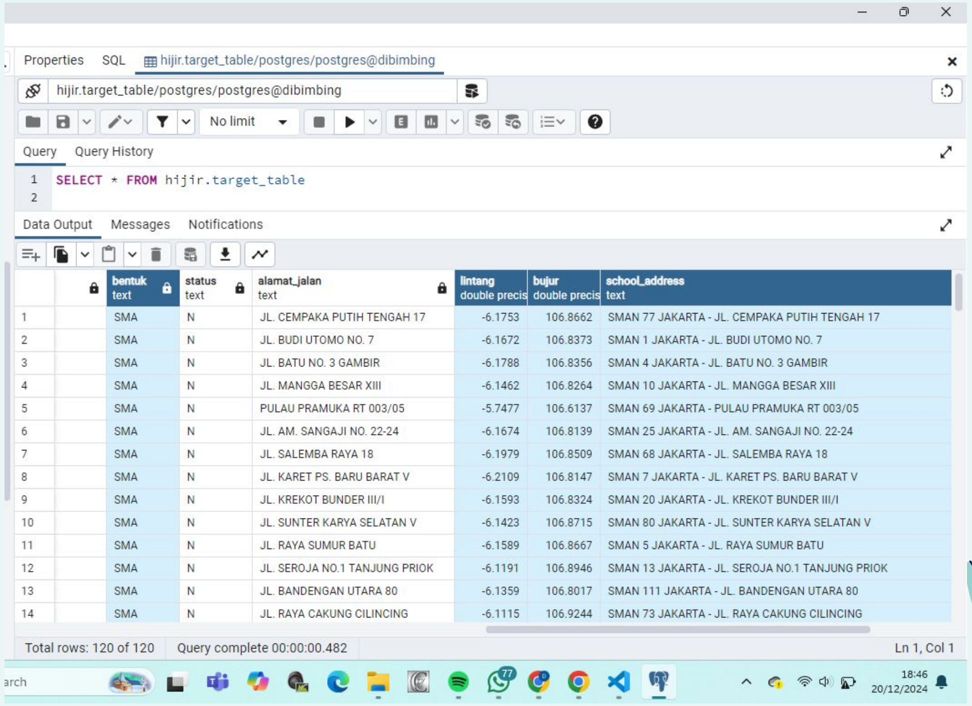


```
[2024-12-20, 11:40:18 UTC] {taskinstance.py:2889} INFO - Executing <Task(PythonOperator): load_data_to_database> on 2024-12-20 11:39:55.410319+00:00
[2024-12-20, 11:40:18 UTC] {warnings.py:112} WARNING - /home/***/.local/lib/python3.12/site-packages/***/task/task_runner/standard_task_runner.py:70: DeprecationWar
  pid = os.fork()
[2024-12-20, 11:40:18 UTC] {standard_task_runner.py:72} INFO - Started process 113333 to run task
[2024-12-20, 11:40:18 UTC] {standard_task_runner.py:104} INFO - Running: ['***', 'tasks', 'run', 'api_to_database_dag_sekolah_hijir', 'load_data_to_database', 'manu
[2024-12-20, 11:40:18 UTC] {standard_task_runner.py:105} INFO - Job 2388: Subtask load_data_to_database
[2024-12-20, 11:40:18 UTC] {task_command.py:467} INFO - Running <TaskInstance: api_to_database_dag_sekolah_hijir.load_data_to_database manual__2024-12-20T11:39:55.
[2024-12-20, 11:40:19 UTC] {taskinstance.py:3132} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='hijir' AIRFLOW_CTX_DAG_ID='api_to_database_dag_sekolah_hijir' A
[2024-12-20, 11:40:19 UTC] {taskinstance.py:731} ▲▲▲ Log group end
[2024-12-20, 11:40:19 UTC] {base.py:84} INFO - Retrieving connection '***'
[2024-12-20, 11:40:19 UTC] {sql.py:544} INFO - Running statement:
    CREATE TABLE IF NOT EXISTS hijir.target_table (
        kode_prop TEXT, propinsi TEXT, kode_kab_kota TEXT, kabupaten_kota TEXT, kode_kec TEXT, kecamatan TEXT, id TEXT, npsn TEXT, sekolah TEXT, bentuk TEXT, statu
    );
    , parameters: None
[2024-12-20, 11:40:19 UTC] {base.py:84} INFO - Retrieving connection '***'
[2024-12-20, 11:40:19 UTC] {python.py:240} INFO - Done. Returned value was: None
[2024-12-20, 11:40:19 UTC] {taskinstance.py:340} ▼ Post task execution logs
[2024-12-20, 11:40:19 UTC] {taskinstance.py:352} INFO - Marking task as SUCCESS. dag_id=api_to_database_dag_sekolah_hijir, task_id=load_data_to_database, run_id=ma
[2024-12-20, 11:40:19 UTC] {local_task_job_runner.py:266} INFO - Task exited with return code 0
[2024-12-20, 11:40:19 UTC] {taskinstance.py:3895} INFO - 0 downstream tasks scheduled from follow-on schedule check
[2024-12-20, 11:40:19 UTC] {local_task_job_runner.py:245} ▲▲▲ Log group end
```

07 Execution and Testing

Testing the Pipeline

1. **Data Accuracy**:
   - Verified data extraction by comparing API response against the raw input in the transformation step.
   - Ensured only SMA schools with status "N" are included in the transformed dataset.
2. **Transformation Validation**:
   - Checked that new columns like `school_address` are correctly generated.
   - Confirmed latitude (`lintang`) and longitude (`bujur`) are numeric and rows with invalid data are removed.
3. **Database Integrity**:
   - Verified table creation in PostgreSQL matches the schema of transformed data.
   - Checked that data is correctly inserted and replaced during each DAG run.
   -


Properties
SQL
hijir.target_table/postgres/postgres@dibimbing
hijir.target_table/postgres/postgres@dibimbing
No limit
Query
Query History

```
SELECT * FROM hijir.target_table
```

Data Output
Messages
Notifications

		bentuk text	status text	alamat_jalan text	lintang double precis	bujur double precis	school_address text
1		SMA	N	JL. CEMPAKA PUTIH TENGAH 17	-6.1753	106.8662	SMAN 77 JAKARTA - JL. CEMPAKA PUTIH TENGAH 17
2		SMA	N	JL. BUDI UTOMO NO. 7	-6.1672	106.8373	SMAN 1 JAKARTA - JL. BUDI UTOMO NO. 7
3		SMA	N	JL. BATU NO. 3 GAMBIR	-6.1788	106.8356	SMAN 4 JAKARTA - JL. BATU NO. 3 GAMBIR
4		SMA	N	JL. MANGGA BESAR XIII	-6.1462	106.8264	SMAN 10 JAKARTA - JL. MANGGA BESAR XIII
5		SMA	N	PULAU PRAMUKA RT 003/05	-5.7477	106.6137	SMAN 69 JAKARTA - PULAU PRAMUKA RT 003/05
6		SMA	N	JL. AM. SANGAJI NO. 22-24	-6.1674	106.8139	SMAN 25 JAKARTA - JL. AM. SANGAJI NO. 22-24
7		SMA	N	JL. SALEMBA RAYA 18	-6.1979	106.8509	SMAN 68 JAKARTA - JL. SALEMBA RAYA 18
8		SMA	N	JL. KARET PS. BARU BARAT V	-6.2109	106.8147	SMAN 7 JAKARTA - JL. KARET PS. BARU BARAT V
9		SMA	N	JL. KREKOT BUNDER III/I	-6.1593	106.8324	SMAN 20 JAKARTA - JL. KREKOT BUNDER III/I
10		SMA	N	JL. SUNTER KARYA SELATAN V	-6.1423	106.8715	SMAN 80 JAKARTA - JL. SUNTER KARYA SELATAN V
11		SMA	N	JL. RAYA SUMUR BATU	-6.1589	106.8667	SMAN 5 JAKARTA - JL. RAYA SUMUR BATU
12		SMA	N	JL. SEROJA NO.1 TANJUNG PRIOK	-6.1191	106.8946	SMAN 13 JAKARTA - JL. SEROJA NO.1 TANJUNG PRIOK
13		SMA	N	JL. BANDENGAN UTARA 80	-6.1359	106.8017	SMAN 111 JAKARTA - JL. BANDENGAN UTARA 80
14		SMA	N	JL. RAYA CAKUNG CILINCING	-6.1115	106.9244	SMAN 73 JAKARTA - JL. RAYA CAKUNG CILINCING

Total rows: 120 of 120   Query complete 00:00:00.482   Ln 1, Col 1

08 Conclusion

1. **ETL Pipeline Overview**:
   - Automated data flow from API to PostgreSQL using Apache Airflow.
   - Tasks: Data extraction, transformation, and loading executed sequentially.
2. **Transformations Applied**:
   - Filtered for **SMA schools** with status "N".
   - Added `school_address` combining name and address.
   - Converted `lintang` and `bujur` to numeric values.
   - Removed rows with invalid or missing coordinates.
3. **Benefits of Apache Airflow**:
   - Simplifies workflows with scheduling, monitoring, and error handling.
   - Scalable and flexible for future data sources or transformations.

Thanks!

Do you have any questions?

hijirdw@gmail.com

https://www.linkedin.com/in/hijirdella/

https://github.com/hijirdella/ETL-Pipeline-Airflow-School-API