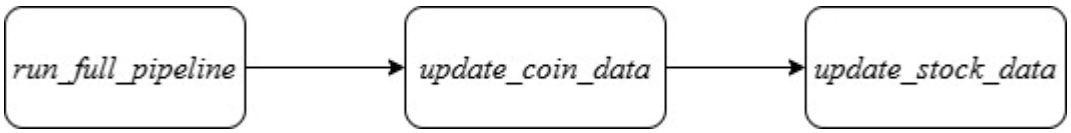
Dokumentasi Komprehensif Airflow - brazil_stock_market_etl

1. Deskripsi DAG

DAG brazil_stock_market_etl bertanggung jawab untuk menjalankan pipeline ETL yang mengolah dan memuat data dari pasar saham Brazil ke dalam data warehouse berbasis DuckDB. Pipeline ini terdiri dari tiga task utama:

1. *run_full_pipeline*
   Menjalankan pipeline ETL penuh untuk inisialisasi data awal (full load). Task ini mencakup ekstraksi data mentah, transformasi, dan load ke dalam data warehouse.
2. *update_coin_data*
   Memperbarui tabel dimensi untuk data koin dari file terbaru ke dalam tabel dimCoin.
3. *update_stock_data*
   Memperbarui tabel dimensi untuk data perusahaan saham ke dalam tabel dimCompany.

2. Diagram Arsitektur & Dependensi Tugas



Setiap task dijalankan menggunakan PythonOperator. Task run_full_pipeline harus berhasil terlebih dahulu sebelum task lainnya dapat diproses. Setelah run_full_pipeline selesai, task update_coin_data dan update_stock_data dapat dijalankan secara paralel atau berurutan, tergantung strategi pencegahan race condition.

3. Deskripsi Tujuan Setiap Tugas

Task Name	Deskripsi
run_full_pipeline	Melakukan full load data awal ke data warehouse.
update_coin_data	Memperbarui data koin dari file CSV ke tabel dimCoin di DuckDB.

Task Name	Deskripsi
update_stock_data	Memperbarui data perusahaan saham dari file CSV ke tabel dimCompany.

4. Penjadwalan (`schedule_interval`)

a. DAG dijalankan **otomatis setiap hari pada pukul 00:00** (`schedule_interval='0 0 * * *'`).
b. `start_date=days_ago(1)` → DAG mulai aktif sejak satu hari sebelum waktu sekarang.
c. `catchup=False` → Airflow hanya menjalankan eksekusi berikutnya, tidak menjalankan eksekusi yang terlewat sejak `start_date`.

Struktur Dependensi Tugas

```
run_full_pipeline → update_coin_data → update_stock_data
```

Penjelasan Masing-Masing Task:

1. `run_full_pipeline`
    - Bertugas untuk menjalankan ETL secara menyeluruh (full load) ke data warehouse.
    - Saat ini, dijadwalkan otomatis setiap hari karena berada dalam DAG harian.
    - Dapat disesuaikan secara manual jika hanya ingin dijalankan sekali saat inisialisasi (misalnya dengan logika kondisi tambahan atau memindahkannya ke DAG terpisah).
2. `update_coin_data`
    - Meng-update tabel `dimCoin` dari file `coin_values.csv`.
    - Berjalan setelah `run_full_pipeline` berhasil.
3. `update_stock_data`
    - Meng-update tabel `dimCompany` dari file `stock_values.csv`.
    - Berjalan setelah `update_coin_data` berhasil.

Catatan Tambahan

1. Semua task dijalankan secara berurutan, bukan paralel, untuk mencegah race condition saat menulis data ke DuckDB.
2. Jika ingin menjadikan `run_full_pipeline` sebagai *initial-only task*, disarankan memisahkannya dari DAG harian.

5. Pengaturan Pemantauan dan Peringatan

A. Monitoring Status:
Seluruh task dapat dipantau melalui UI Airflow dalam berbagai tampilan, antara lain:

1. Tree View – Menampilkan status eksekusi setiap task berdasarkan waktu
2. Graph View – Menunjukkan alur dependensi antar task dalam DAG
3. Log per Task – Menyediakan detail eksekusi dan error log untuk debugging

B. Peringatan (Alerting) – Opsional:

1. Jika diperlukan, sistem dapat dikonfigurasi menggunakan `EmailOperator` atau webhook (misalnya Slack atau Discord) untuk mengirim notifikasi apabila terjadi kegagalan pada task.
2. Untuk kebutuhan skala besar, Airflow dapat diintegrasikan dengan sistem monitoring eksternal seperti Grafana melalui Prometheus + Airflow Exporter.

6. Prosedur Pemulihan Kegagalan

1. *Identifikasi kesalahan:*
    - Buka log task yang gagal dari UI Airflow.
    - Analisis pesan error dan trace log.
2. *Perbaiki penyebab kegagalan:*
    - Misalnya: periksa path file, koneksi database, atau format data CSV.
3. *Lakukan re-run task*:
    - Pilih task yang gagal, klik "Clear", lalu klik tombol “Run” ulang di Airflow.
    - Untuk daily task, cukup re-run task tersebut tanpa perlu mengulang run_full_pipeline.
4. *Pastikan status berubah menjadi "Success"*.