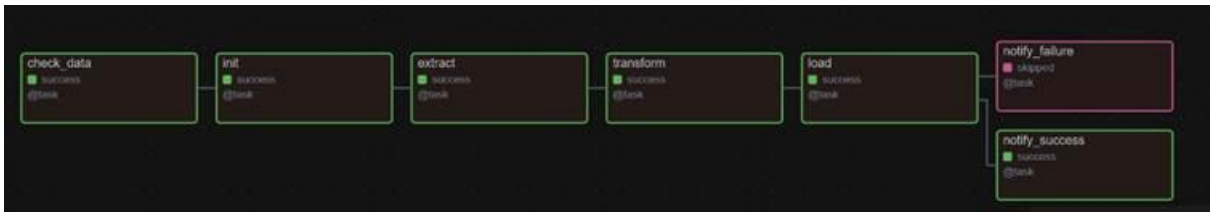
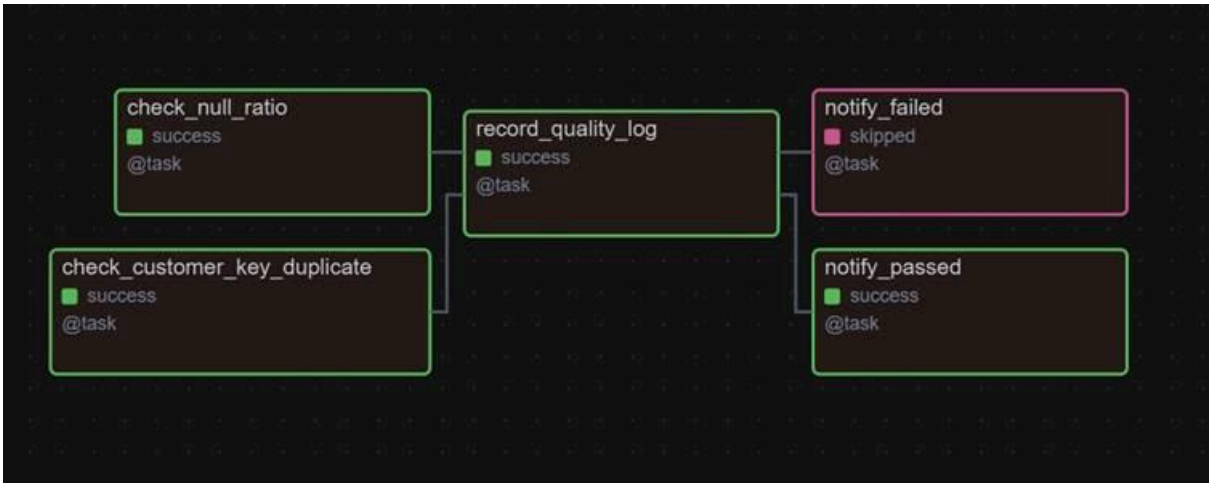
DOKUMENTASI IMPLEMENTASI AIRFLOW

A. Diagram arsitektur



Gambar 1.1 Diagram arsitektur etl_duckdb_dag.py



Gambar 1.2 Diagram arsitektur Dag_data_quality.py

1. Deskripsi tujuan dari setiap tugas

1.1 Diagram arsitektur etl_duckdb_dag.py

Proses ETL diatur dalam sebuah DAG (Directed Acyclic Graph) yang terdiri atas rangkaian tugas (task) yang dijalankan secara berurutan. DAG ini mencerminkan urutan logis proses ETL:

1.1.1 check_data : Memastikan ketersediaan file data mentah (data.csv) dari Google Drive atau lokal.

1.1.2 init: Menginisialisasi struktur tabel di dalam DuckDB, mencakup tabel dimensi (dim_customer, dim_card, dll.) dan tabel fakta (fact_transactions).

1.1.3 extract: Mengekstrak data dari file CSV.

1.1.4 transform: Mengolah data mentah menjadi bentuk tabel dimensi dan fakta yang telah distandarkan.

1.1.5 load: Memasukkan data hasil transformasi ke warehouse DuckDB.

Setelah proses utama selesai, sistem secara otomatis menentukan jalur notifikasi:

- notify_success akan mengirim email ke stakeholder jika seluruh proses berhasil.
- notify_failure akan aktif jika terjadi kegagalan pada salah satu task dalam pipeline.

1.2 Diagram arsitektur Dag_data_quality.py

DAG ini terdiri dari beberapa task penting yang berjalan secara paralel dan dilanjutkan ke pencatatan dan pengiriman notifikasi. Task-task tersebut adalah:

1.2.1 check_null_ratio : Task ini bertugas menghitung rasio nilai null di setiap kolom dalam tabel fakta fact_transactions. Jika rasio nilai null pada suatu kolom melebihi ambang batas tertentu (misalnya 10%), task ini akan gagal dan memicu mekanisme peringatan otomatis.

1.2.2 check_customer_key_duplicate : Fokus pada integritas data di tabel dimensi dim_customer, task ini memastikan bahwa customer_key bersifat unik dan tidak terjadi duplikasi. Duplikasi kunci pelanggan bisa menyebabkan kesalahan dalam proses join atau analisis downstream.

1.2.3 record_quality_log : Setelah dua pemeriksaan di atas berhasil, hasilnya dicatat dalam log kualitas data. Log ini dapat disimpan dalam bentuk file CSV atau format lainnya, dan menjadi bahan monitoring atau audit historis dari validitas data setiap harinya.

1.2.4 notify_passed dan notify_failed : Dua task ini bertugas mengirimkan notifikasi email kepada pihak terkait. notify_passed akan dikirimkan apabila semua pemeriksaan berhasil. Sebaliknya, notify_failed akan dikirim jika salah satu pemeriksaan gagal.

Evaluasi Hasil Eksekusi : Berdasarkan visualisasi DAG yang ditampilkan, dapat disimpulkan bahwa semua task utama telah berjalan dengan sukses. Status success berwarna hijau pada semua task menunjukkan bahwa:

- Tidak ada kolom dengan nilai null melebihi threshold
- Tidak ditemukan duplikasi pada customer_key
- Log kualitas data telah berhasil dicatat
- Notifikasi keberhasilan berhasil dikirimkan
- Task notifikasi kegagalan notify_failed otomatis di-skip karena tidak relevan

Hal ini menandakan bahwa kualitas data saat ini berada dalam kondisi yang baik dan dapat digunakan dengan aman untuk analisis atau visualisasi lebih lanjut.

2. **Informasi penjadwalan dan dependensi**

   DAG dijadwalkan untuk berjalan setiap hari (@daily) dengan pengaturan catchup=False, artinya hanya akan mengeksekusi untuk hari-hari berjalan dan tidak akan memproses data backlog. Ketergantungan antar task didefinisikan secara eksplisit menggunakan TaskFlow API dan operator >>.

3. **Pengaturan pemantauan dan peringatan**

   Pengawasan sistem dilakukan dengan pendekatan kombinasi:

   4.1 Airflow Web UI memberikan gambaran visual dari alur dan status masing-masing task.

   4.2 Sistem email notifikasi dikonfigurasi untuk mengirim peringatan ke tiga akun email UGM jika terjadi kegagalan, serta konfirmasi keberhasilan proses.

   DAG tambahan **dag_data_quality** digunakan untuk memantau kualitas data, mengevaluasi duplikasi customer_key, dan rasio nilai null, serta mencatat hasil monitoring ke dalam file **log_data_quality.csv.**

4. **Prosedur pemulihan kegagalan**

   Jika terjadi kegagalan pada salah satu task, Airflow akan:

   a. Menandai task sebagai gagal,

   b. Menjalankan task notify_failure untuk memberi tahu tim melalui email,

   c. Melakukan retry otomatis jika sudah dikonfigurasi (retries),

   d. Memberikan akses ke log kesalahan yang dapat diperiksa melalui UI Airflow.

B. Strategi Penjadwalan

Dalam memenuhi kebutuhan proyek ini, penjadwalan DAG kami rancang dengan mempertimbangkan karakteristik data harian dan kebutuhan notifikasi yang responsif. Oleh karena itu, dua DAG utama yaitu etl_duckdb_dag_custom_class dan dag_data_quality dijalankan secara otomatis setiap hari menggunakan parameter schedule_interval='@daily'. Pemilihan frekuensi harian ini didasarkan pada asumsi bahwa data transaksi dan atribut pelanggan

diperbarui setiap 24 jam. Ini juga memberikan waktu yang cukup untuk pemrosesan, validasi, dan intervensi manual bila diperlukan.

Selain itu, penggunaan start_date=days_ago(1) memastikan bahwa DAG langsung aktif tanpa perlu mengatur tanggal tertentu secara eksplisit. Konfigurasi catchup=False digunakan agar Airflow tidak menjalankan DAG untuk hari-hari sebelumnya jika sempat terlewat. Ini penting agar pipeline tidak terbebani oleh backlog yang tidak relevan dalam konteks data real-time atau near real-time.

Strategi dependensi antar task juga ditentukan secara eksplisit menggunakan operator >> dan TriggerRule. Task notifikasi sukses (notify_passed) hanya akan berjalan jika semua task sebelumnya berhasil (ALL_SUCCESS), sedangkan notifikasi kegagalan (notify_failed) akan dipicu jika ada satu task saja yang gagal (ONE_FAILED).

Dengan strategi ini, sistem penjadwalan tidak hanya menjamin data terbaru dapat diproses dan divalidasi secara otomatis, tetapi juga memberikan kontrol penuh kepada tim data untuk memantau dan menangani kasus kegagalan secara proaktif melalui sistem peringatan berbasis email.

C. Metrik Kualitas Data

Dalam membuat metrik kualitas data yang baik, pipeline ini kami lengkapi dengan mekanisme pemeriksaan kualitas data yang dijalankan melalui DAG khusus bernama dag_data_quality.

Pemeriksaan ini dilakukan berdasarkan dua metrik utama: duplikasi kunci utama (primary key duplication) dan rasio nilai kosong (null ratio). Fokus utama adalah pada tabel dim_customer dan fact_transactions, yang merupakan inti dari struktur data warehouse.

Metrik pertama yang digunakan adalah pemeriksaan duplikasi pada kolom customer_key di tabel dim_customer. Kunci utama seharusnya bersifat unik untuk menjamin integritas referensial. Jika ditemukan nilai customer_key yang muncul lebih dari sekali, maka task dianggap gagal, karena hal ini berpotensi menyebabkan duplikasi dalam hasil join dan merusak keakuratan laporan.

Metrik kedua adalah penghitungan rasio nilai null di setiap kolom pada tabel fact_transactions. Sebuah ambang batas (threshold) sebesar 10% diterapkan. Jika suatu kolom memiliki lebih dari 10% nilai kosong, maka dianggap gagal. Rasio ini dipilih karena dianggap masih toleran untuk kebutuhan analisis deskriptif, namun cukup ketat untuk menjaga validitas pada laporan operasional dan dasbor interaktif.

Setelah pemeriksaan dilakukan, hasilnya dicatat dalam file log harian (log_data_quality.csv) yang menyimpan informasi seperti tanggal, status sukses/gagal, serta detail nilai metrik. Jika semua metrik berhasil dilewati, sistem akan mengirimkan email notifikasi keberhasilan kepada tim yang ditentukan. Sebaliknya, jika salah satu metrik gagal, task notify_failed akan mengirimkan email peringatan disertai pesan kesalahan.

Pendekatan berbasis metrik ini tidak hanya memberikan pengawasan otomatis terhadap kualitas data, tetapi juga memungkinkan pencatatan historis dan evaluasi tren kualitas dari waktu ke waktu. Dengan dokumentasi ini, proyek ETL tidak hanya fokus pada pemrosesan data, tetapi juga menjamin akurasi dan keandalan data sebagai aset organisasi.