

Sales Performance Analysis

2020-2023

Kimia Farma - Big Data Analytics

Presented by:
Shifly Taysir Setiawan



Daftar Isi

1. Profile
2. Company Overview
3. Project Portfolio
4. Importing Dataset into BigQuery
5. Creating Analysis Table
6. BigQuery Syntax
7. Dashboard Performance Analytics





Shifly Taysir Setiawan

Surabaya | linkedin.com/in/shifly/ | shiflytaysirsetiawan@gmail.com

Saya adalah seorang mahasiswa Sistem Informasi di Institut Teknologi Sepuluh Nopember (ITS). Saya memiliki fokus keahlian dalam desain grafis dan visualisasi data. Untuk mendukung kemampuan tersebut, saya juga memiliki kompetensi teknikal dalam programming menggunakan Laravel, PHP, SQL, dan Python.

Pengalaman saya hingga saat ini:

- Information Media Staff - HMSI ITS (Himpunan Mahasiswa Sistem Informasi ITS)
- Brand Designer - 180 Degrees Consulting ITS
- IS Administrator - Apotek Shifa
- Staff of Data Management - Information Systems Expo (ISE!) 2024
- Merchandise Manager - Petroleum Integrated Days (Petrolida) 2025

Company Overview

PT Kimia Farma Tbk adalah perusahaan farmasi pertama di Indonesia (berdiri sejak 1817) yang telah berkembang menjadi penyedia layanan kesehatan terintegrasi terkemuka. Perusahaan ini menjadi perusahaan publik (Tbk) pada tahun 2001 dan, sejak tahun 2020, resmi menjadi anggota Holding BUMN Farmasi dengan PT Bio Farma (Persero) sebagai induk perusahaan setelah mengambil alih mayoritas saham milik pemerintah.

Selama perjalannya, Kimia Farma telah bertransformasi menjadi salah satu penyedia layanan kesehatan terintegrasi terkemuka di Indonesia.



Project Portofolio

Tugas saya adalah berperan sebagai Big Data Analytics Intern untuk mengevaluasi kinerja bisnis Kimia Farma dari tahun 2020-2023. Proyek ini mencakup pemrosesan data di BigQuery dan pembuatan dasbor interaktif di Looker Studio untuk menganalisis tren penjualan, profitabilitas, dan kinerja cabang

Dalam project ini saya menggunakan tools berikut:



Google
Big Query



Looker Studio



GitHub

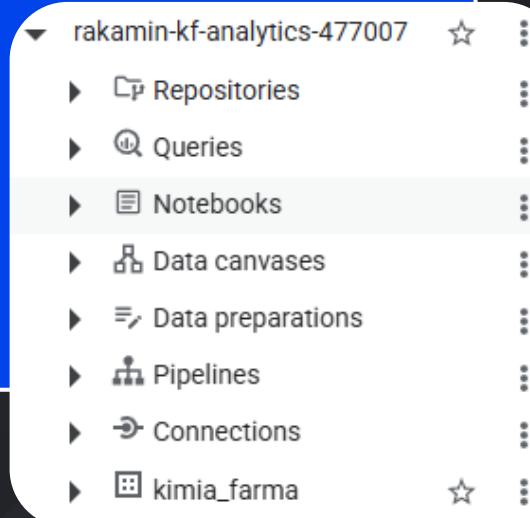
Importing Dataset → BigQuery

Strategi untuk tahap ini adalah membangun fondasi data yang bersih dan terstruktur di Google Cloud BigQuery.

Ini melibatkan tiga langkah kunci:

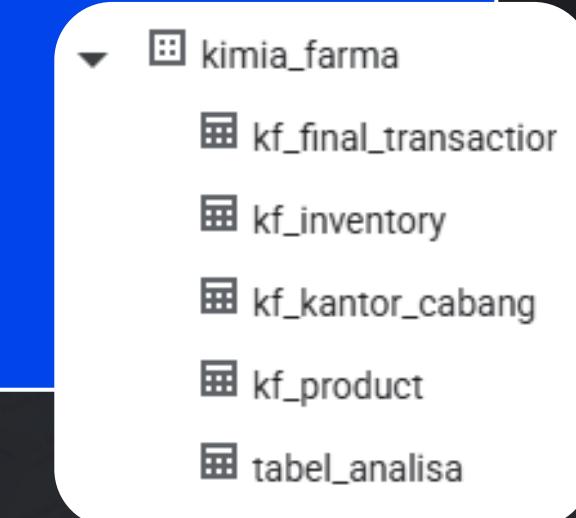
1 Setup Lingkungan

Langkah pertama adalah membuat project baru dan dataset khusus (kimia_farma) . Ini berfungsi sebagai 'wadah' terisolasi untuk semua data terkait proyek.



2 Proses Impor Data

Keempat file CSV yang disediakan (kf_final_transaction, kf_kantor_cabang, kf_product, kf_inventory) diimpor satu per satu ke dalam BigQuery.



3 Deteksi Skema

Selama proses impor, saya memanfaatkan fitur "Auto detect schema". Strategi ini sangat penting untuk memastikan BigQuery secara otomatis mengenali tipe data yang benar untuk setiap kolom (misalnya, price sebagai Angka, date sebagai Teks/String, rating sebagai Angka Desimal). Ini mencegah kesalahan data sebelum kueri SQL dijalankan.

Creating Analysis Table

Untuk analisis, data mentah yang terpisah tidak efisien. Oleh karena itu, satu tabel utama (tabel_analisa) dibuat dengan menggabungkan tabel-tabel tersebut menggunakan kueri SQL. Selama proses JOIN, kolom-kolom kalkulasi baru dibuat, seperti nett_sales (harga - diskon) , persentase_gross_laba (menggunakan logika CASE...WHEN berdasarkan harga) , dan nett_profit.

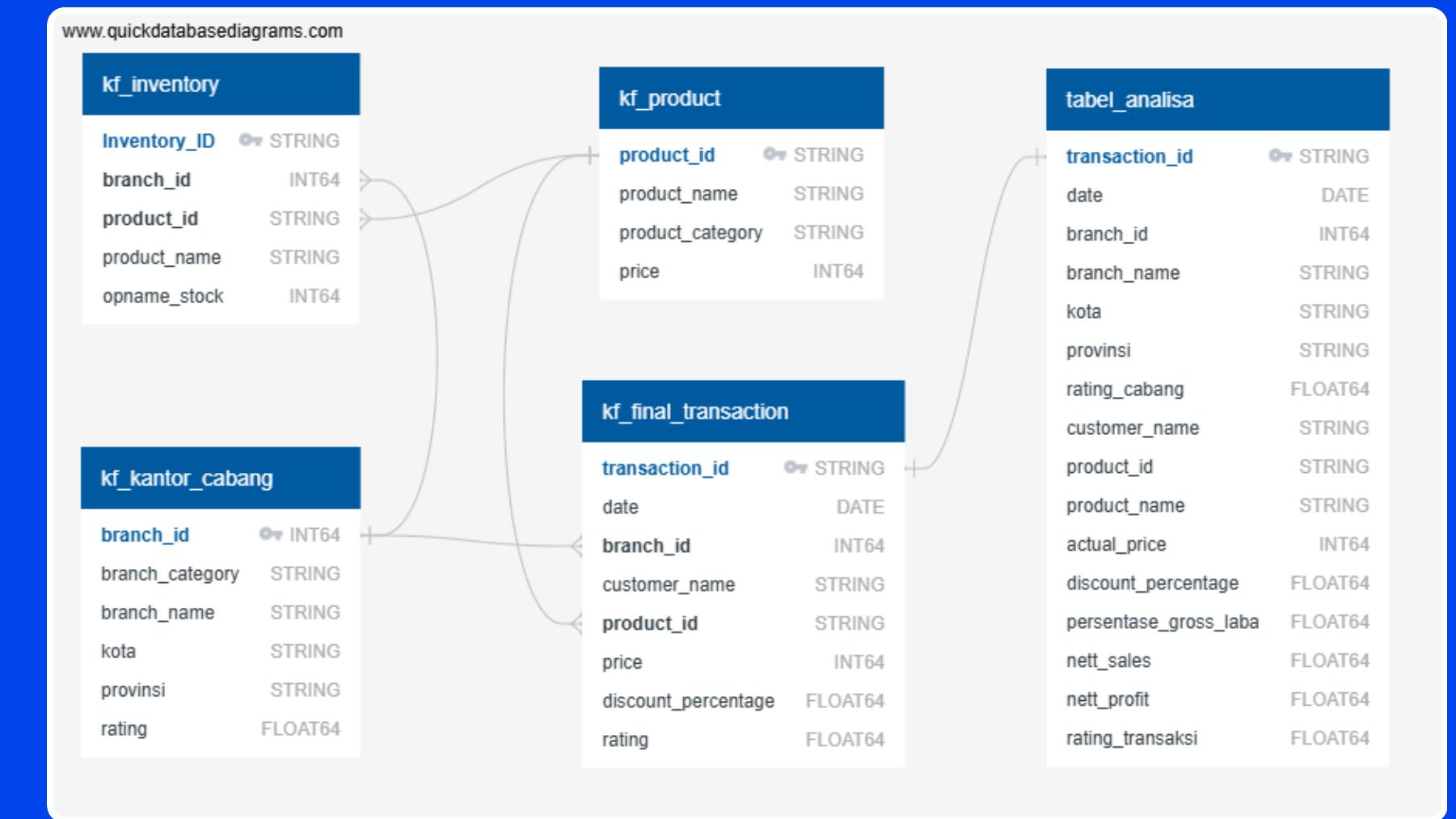


Diagram ERD ini memvisualisasikan arsitektur data. Tabel kf_product dan kf_kantor_cabang bertindak sebagai tabel induk (dimensi). Tabel kf_final_transaction dan kf_inventory adalah tabel anak (fakta) yang terhubung ke kedua induk. Terakhir, tabel_analisa (di bagian bawah) adalah hasil denormalisasi yang menjadi sumber data tunggal untuk dasbor.

BigQuery Syntax - 1

Bagian pertama query ini menggunakan Common Table Expression (CTE) bernama BaseJoin. Tujuannya adalah untuk mengumpulkan semua data mentah yang diperlukan ke dalam satu tampilan sementara. Ini dilakukan dengan mengambil kf_final_transaction sebagai tabel utama, lalu menggabungkannya (LEFT JOIN) dengan kf_kantor_cabang menggunakan branch_id, dan menggabungkannya lagi dengan kf_product menggunakan product_id.

Hasilnya adalah tabel virtual yang berisi semua kolom yang kita butuhkan untuk perhitungan di langkah selanjutnya.

```
1 CREATE TABLE `rakamin-kf-analytics-477007.kimia_farma.tabel_analisa` AS
2
3 WITH
4   -- JOIN TABLE
5   BaseJoin AS (
6     SELECT
7       t.transaction_id,
8       t.date AS date,
9       t.branch_id,
10      t.customer_name,
11      t.product_id,
12      t.price,
13      t.discount_percentage,
14      t.rating AS rating_transaksi,
15      c.branch_name,
16      c.kota,
17      c.provinsi,
18      c.rating AS rating_cabang,
19      p.product_name
20   FROM
21     `rakamin-kf-analytics-477007.kimia_farma.kf_final_transaction` AS t
22   LEFT JOIN
23     `rakamin-kf-analytics-477007.kimia_farma.kf_kantor_cabang` AS c
24   ON t.branch_id = c.branch_id
25   LEFT JOIN
26     `rakamin-kf-analytics-477007.kimia_farma.kf_product` AS p
27   ON t.product_id = p.product_id
28 ).
```

BigQuery Syntax - 2

Bagian kedua, CTE Calculations, mengambil data dari BaseJoin dan mulai menghitung metrik bisnis. Di sinilah logika bisnis dari instruksi diterapkan. Pertama, kolom persentase_gross_laba dibuat menggunakan pernyataan CASE & WHEN untuk menetapkan margin laba yang berbeda berdasarkan rentang harga produk .

Kedua, kolom nett_sales dihitung dengan mengurangkan diskon dari harga ($price - (price * discount_percentage)$) untuk mendapatkan pendapatan bersih per transaksi.

```
30  -- HITUNG KOLOM TAMBAHAN
31  Calculations AS (
32    SELECT
33      transaction_id,
34      date,
35      branch_id,
36      customer_name,
37      product_id,
38      price,
39      discount_percentage,
40      rating_transaksi,
41      branch_name,
42      kota,
43      provinsi,
44      rating_cabang,
45      product_name,
46
47  -- HITUNG PERSENTASE LABA
48  CASE
49    WHEN price <= 50000 THEN 0.10
50    WHEN price > 50000 AND price <= 100000 THEN 0.15
51    WHEN price > 100000 AND price <= 300000 THEN 0.20
52    WHEN price > 300000 AND price <= 500000 THEN 0.25
53    WHEN price > 500000 THEN 0.30
54  END AS persentase_gross_laba,
55
56  -- HITUNG NETT SALES
57  (price - (price * discount_percentage)) AS nett_sales
58
59  FROM BaseJoin
60 )
```

BigQuery Syntax - 3

Bagian terakhir adalah SELECT final yang mengambil data dari CTE Calculations dan membentuk tabel_analisa yang baru. Di sinilah perhitungan terakhir, nett_profit, dibuat dengan mengalikan nett_sales dengan persentase_gross_laba yang sudah dihitung.

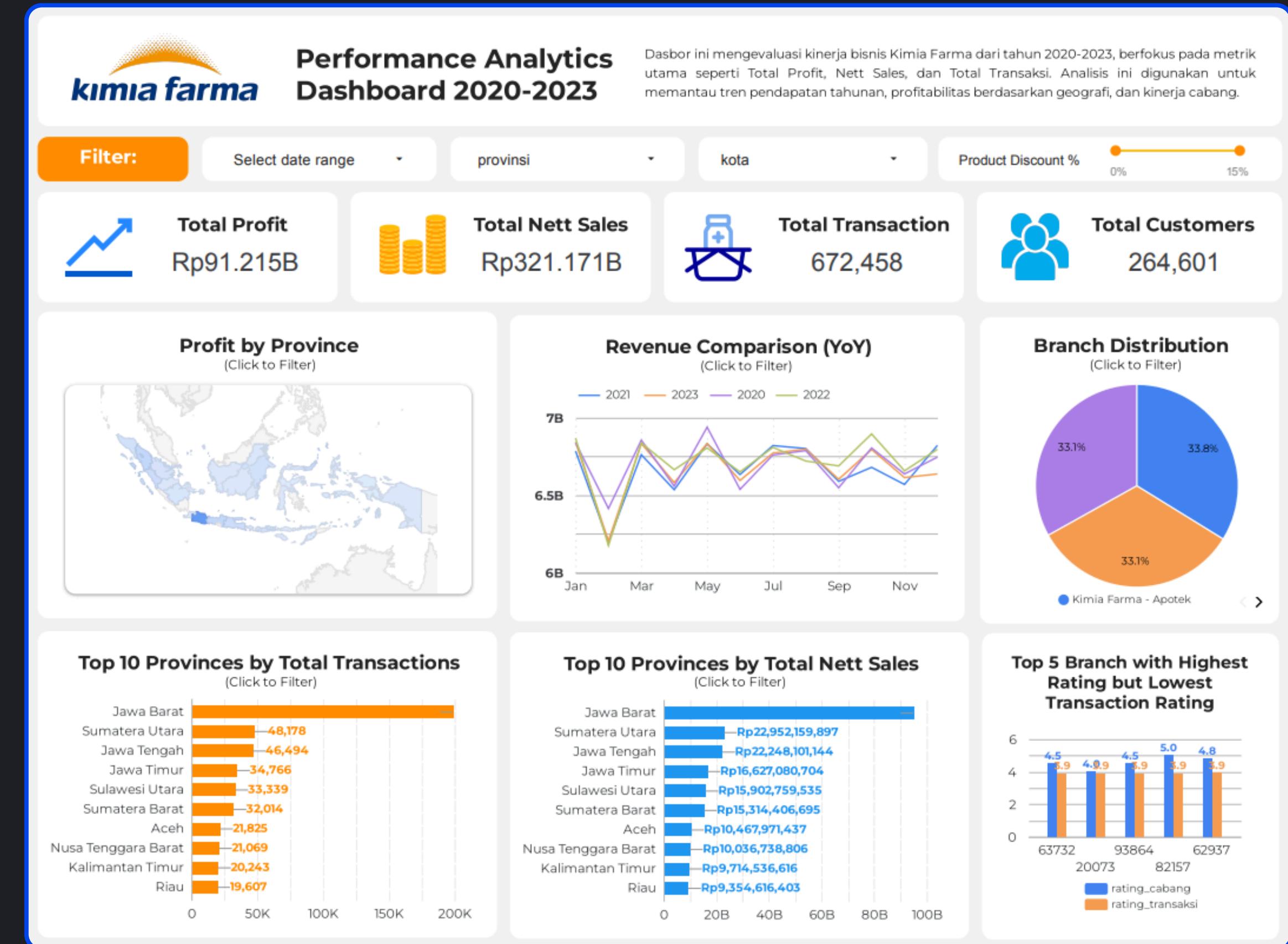
Perintah ini juga memilih semua kolom akhir sesuai urutan yang diminta, mengganti nama price menjadi actual_price, dan secara fisik membuat tabel baru di BigQuery sebagai sumber data untuk dasbor Looker Studio.

```
62 -- KOLOM AKHIR
63 SELECT
64   transaction_id,
65   date,
66   branch_id,
67   branch_name,
68   kota,
69   provinsi,
70   rating_cabang,
71   customer_name,
72   product_id,
73   product_name,
74   price AS actual_price,
75   discount_percentage,
76   persentase_gross_laba,
77   nett_sales,
78   (nett_sales * persentase_gross_laba) AS nett_profit,
79   rating_transaksi
80 FROM
81   Calculations;
```

Dashboard Performance Analytics

Dasbor Google Looker Studio yang interaktif ini terhubung langsung ke tabel_analisa di Google BigQuery.

Dasbor ini dirancang untuk memberikan gambaran lengkap kinerja bisnis dalam satu layar, dilengkapi dengan filter tanggal dan provinsi untuk eksplorasi data.



Important Links

Github: <https://github.com/shiflytaysir/Rakamin-Big-Data-Analyst-KF>

Looker: <https://lookerstudio.google.com/reporting/ac747559-2fe7-4cb8-8c7b-53e547a46e9e>

Explanation Video: -



Thank
You

Presented by:
Shifly Taysir Setiawan