**DOKUMEN QUERY ANALYZING ECOMMERCE BUSINESS PERFORMANCE WITH SQL**

**Disusun Oleh :**

**Ammar Khaq Baasir**

**Linkedin :** [**https://www.linkedin.com/in/ammar-baasir-973873234/**](https://www.linkedin.com/in/ammar-baasir-973873234/)

**Lihat & Download File Format SQL :** [**https://drive.google.com/file/d/1bFhxOQ5sOdBvrUtu7mMB230jRxNdZ5ds/view?usp=sharing**](https://drive.google.com/file/d/1bFhxOQ5sOdBvrUtu7mMB230jRxNdZ5ds/view?usp=sharing)

**DAFTAR ISI**

Task 1 :

**Data Preprocessing**

Task 2 :

**Annual Customer Activity Growth Analysis**

Tasl 3 :

**Annual Product Category Quality Analysis**

Task 4 :

**Analysis of Annual Payment Type Usage**

**TASK 1**

Part 1 :

Membuat sebuah database baru beserta tabel – tabelnya dan menentukan jenis tipe data yang digunakan pada setiap kolomnya.

* Query Pembuatan Database :

CREATE DATABASE Mini\_project1;

* Query Pembuatan Tabel, Mendefiniskan Tipe Data, dan Memberikan Dokumentasi berupa keterangan setiap kolom pada setiap tabel.

CREATE TABLE IF NOT EXISTS public.customers\_dataset

(

    customer\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    customer\_unique\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    customer\_zip\_code\_prefix numeric(6, 0) NOT NULL,

    customer\_city character varying(50) COLLATE pg\_catalog."default" NOT NULL,

    customer\_state character varying(5) COLLATE pg\_catalog."default" NOT NULL,

    CONSTRAINT customers\_dataset\_pkey PRIMARY KEY (customer\_id),

    CONSTRAINT customers\_dataset\_customer\_id\_customer\_id1\_key UNIQUE (customer\_id)

        INCLUDE(customer\_id)

);

COMMENT ON TABLE public.customers\_dataset

    IS 'tabel yang menyimpan data dari customers :

- customer\_id : id dari customers,

- customer\_unique\_id : unique id dari customers,

- customer\_zip\_code\_prefix : kode pos tempat tinggal dari customers,

- customer\_city : nama kota dari customers,

- customer\_state  nama state (kode negara bagian) tempat tinggal dari customers: ';

CREATE TABLE IF NOT EXISTS public.geolocations\_dataset

(

    geolocation\_zip\_code\_prefix numeric(6, 0) NOT NULL,

    geolocation\_lat numeric(20, 0) NOT NULL,

    "geolocation\_Ing" numeric(20, 0) NOT NULL,

    geolocation\_city character varying(50) COLLATE pg\_catalog."default" NOT NULL,

    geolocation\_state character varying(6) COLLATE pg\_catalog."default" NOT NULL

);

COMMENT ON TABLE public.geolocations\_dataset

    IS 'tabel yang menyimpan data geolocations :

- geolocation\_zip\_code\_prefix : menunjukkan kode pos dari lokasi geografis,

- geolocation\_lat : menunjukkan posisi lintang(latitude) dari lokasi geografis,

- geolocation\_lng : menunjukkan posisi bujur(longitude) dari lokasi geografis,

- geolocation\_city : menunjukkan nama kota dari lokasi geografis,

- geolocation\_state : menunjukkan nama (kode nama) dari state (negara bagian)';

CREATE TABLE IF NOT EXISTS public.order\_items\_dataset

(

    order\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    order\_item\_id numeric(3, 0) NOT NULL,

    product\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    seller\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    shipping\_limit\_date timestamp with time zone NOT NULL,

    price double precision NOT NULL,

    freight\_value double precision NOT NULL

);

COMMENT ON TABLE public.order\_items\_dataset

    IS 'tabel ini menyimpan data order atau transaksi dari customers :

- order\_id : menunjukkan id order,

- order\_item\_id : menunjukkan banyaknya item yang di order,

- product\_id : menunjukkan id product,

- seller\_id : menunjukkan id sellers,

- shipping\_limit\_date : menunjukkan batas waktu pengiriman product (pesanan) dimana batasan waktu yang diberikan kepada sellers yang harus menyerahkan product pada ekspedisi pengiriman,

- price : menunjukkan harga dari product,

- freight\_value : menunjukkan biaya pengiriman (ongkos kirim)';

CREATE TABLE IF NOT EXISTS public.orders\_dataset

(

    order\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    customer\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    order\_status character varying(12) COLLATE pg\_catalog."default" NOT NULL,

    order\_purchase\_timestamp timestamp with time zone NOT NULL,

    order\_approved\_at timestamp with time zone,

    order\_delivered\_carrier\_date timestamp with time zone,

    order\_delivered\_customer\_date timestamp with time zone,

    order\_estimated\_delivery\_date timestamp with time zone,

    CONSTRAINT orders\_dataset\_pkey PRIMARY KEY (order\_id),

    CONSTRAINT orders\_dataset\_order\_id\_order\_id1\_key UNIQUE (order\_id)

        INCLUDE(order\_id)

);

COMMENT ON TABLE public.orders\_dataset

    IS 'tabel ini berisikan data order dari customers :

order\_id : menunjukkan id order,

customer\_id : menunjukkan id customer,

order\_status : menunjukkan status dari order,

order\_purchase\_timestamp : menunjukkan waktu pertama kali customer melakukan order,

order\_approved\_at : menunjukkan waktu pesanan dari customer disetujui oleh sistem,

order\_delivered\_carrier\_date : menunjukkan waktu dimana pesanan customer diterima ekspedisi pengiriman.

order\_delivered\_customer\_date : menunjukkan waktu dimana pesanan telah diterima customer (pihak ekspedisi pengiriman menyerahkan kepada customer).

order\_estimated\_delivery\_date : menunjukkan waktu dimana estimasi awal pesanan yang diterima customer (lama waktu dari customer memesan hingga customer menerima pesanannya)';

CREATE TABLE IF NOT EXISTS public.payments\_dataset

(

    order\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    payment\_sequential numeric(3, 0) NOT NULL,

    payment\_type character varying(15) COLLATE pg\_catalog."default" NOT NULL,

    payment\_installments numeric(2, 0) NOT NULL,

    payment\_value numeric(5, 0) NOT NULL

);

COMMENT ON TABLE public.payments\_dataset

    IS 'tabel ini berisikan data payments dari customers :

- order\_id : menunjukkan id order,

- payment\_sequential : menunjukkan jumlah atau banyaknya pembayaran yang dilakukan dalam satu pesanan,

- payment\_type : menunjukkan jenis tipe pembayaran yang digunakan,

- payment\_installments : menunjukkan banyaknya (berapa kali) pembayaran yang dilakukan ini dapat berisi banyaknya jumlah angsuran,

- payment\_value : menunjukkan jumlah besaran pembayaran yang dilakukan';

CREATE TABLE IF NOT EXISTS public.products\_dataset

(

    product\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    product\_category\_name character varying(50) COLLATE pg\_catalog."default",

    product\_name\_lenght double precision,

    product\_description\_lenght double precision,

    product\_photos\_qty double precision,

    product\_weight\_g double precision,

    product\_length\_cm double precision,

    product\_height\_cm double precision,

    product\_width\_cm double precision,

    CONSTRAINT products\_dataset\_pkey PRIMARY KEY (product\_id),

    CONSTRAINT products\_dataset\_product\_id\_product\_id1\_key UNIQUE (product\_id)

        INCLUDE(product\_id)

);

COMMENT ON TABLE public.products\_dataset

    IS 'tabel ini menyimpan data dari jenis - jenis product :

- product\_id : menunjukkan id product

- product\_category\_name : menunjukkan nama kategori dari product,

- product\_name\_lenght : menunjukkan panjang nama product,

- product\_description\_lenght : menunjukkan panjang deskripsi dari product,

- product\_photos\_qty : menunjukkan gambar quantity dati product

- product\_weight\_g : menunjukkan ukuran berat product berdasarkan gram

- product\_length\_cm : menunjukkan ukuran panjang product berdasarkan cm

- product\_height\_cm : menunjukkan ukuran tinggi product berdasarkan cm

- product\_width\_cm : menunjukkan ukuran lebar product berdasarkan cm';

CREATE TABLE IF NOT EXISTS public.reviews\_dataset

(

    review\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    order\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    review\_score numeric(1, 0) NOT NULL,

    review\_comment\_title character varying(50) COLLATE pg\_catalog."default",

    review\_comment\_message character varying(350) COLLATE pg\_catalog."default",

    review\_creation\_date timestamp with time zone NOT NULL,

    review\_answer\_timestamp timestamp with time zone NOT NULL

);

COMMENT ON TABLE public.reviews\_dataset

    IS 'tabel ini menyimpan data reviews dari customers :

- review\_id : menunjukkan id review,

- order\_id : menunjukkan id order,

- review\_score : menunjukkan nilai review dari customer,

- review\_comment\_title : menunjukkan judul review dari customer

- review\_comment\_message : menunjukkan isi review dari customer

- review\_creation\_date : menunjukkan waktu riview yang dibuat atau diberikan oleh customer

- review\_answer\_timestamp : menunjukkan waktu dari jawaban (feedback) yang diberikan atas review dari customer';

CREATE TABLE IF NOT EXISTS public.sellers\_dataset

(

    seller\_id character varying(36) COLLATE pg\_catalog."default" NOT NULL,

    seller\_zip\_code\_prefix numeric(6, 0) NOT NULL,

    seller\_city character varying(50) COLLATE pg\_catalog."default" NOT NULL,

    seller\_state character varying(5) COLLATE pg\_catalog."default" NOT NULL,

    CONSTRAINT sellers\_dataset\_pkey PRIMARY KEY (seller\_id),

    CONSTRAINT sellers\_dataset\_seller\_id\_seller\_id1\_key UNIQUE (seller\_id)

        INCLUDE(seller\_id)

);

COMMENT ON TABLE public.sellers\_dataset

    IS 'tabel yang menyimpan data dari sellers :

- seller\_id : menunjukkan id unique dari sellers,

- seller\_zip\_code\_prefix : menunjukkan kode pos tempat tinggal sellers,

- seller\_city : menunjukkan nama kota tempat tinggal sellers,

- seller\_state : menunjukkan nama state (kode negara bagian) tempat tinggal sellers';

ALTER TABLE IF EXISTS public.order\_items\_dataset

    ADD CONSTRAINT order\_items\_dataset\_order\_id\_fkey FOREIGN KEY (order\_id)

    REFERENCES public.orders\_dataset (order\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.order\_items\_dataset

    ADD CONSTRAINT order\_items\_dataset\_product\_id\_fkey FOREIGN KEY (product\_id)

    REFERENCES public.products\_dataset (product\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.order\_items\_dataset

    ADD CONSTRAINT order\_items\_dataset\_seller\_id\_fkey FOREIGN KEY (seller\_id)

    REFERENCES public.sellers\_dataset (seller\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.orders\_dataset

    ADD CONSTRAINT orders\_dataset\_customer\_id\_fkey FOREIGN KEY (customer\_id)

    REFERENCES public.customers\_dataset (customer\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public.payments\_dataset

    ADD CONSTRAINT payments\_dataset\_order\_id\_fkey FOREIGN KEY (order\_id)

    REFERENCES public.orders\_dataset (order\_id) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

END;

Pada database Mini\_project1 ini terdiri atas 8 tabel yang telah didefinisikan pada query diatas. 8 tabel yang dibuat menyimpan informasi yang berbeda – beda, yakni sebagai berikut :

-   customers\_dataset :

tabel ini menyimpan informasi data – data dari customers yang melakukan pembelian pada ecommerce. Data – data customers ini berupa Alamat, customers id, city, dll.

-   geolocations\_dataset :

tabel yang menyimpan informasi data geolocation dari customers maupun sellers. Data geolocation berupa garis lintang (latitude, longitude).

-   order\_items\_dataset :

tabel yang menyimpan informasi mengenai data product yang dibeli oleh customers. Data yang tersimpan pada tabel ini berupa data harga product, ongkos kirim, dll.

-   orders\_dataset :

tabel yang menyimpan informasi mengenai informasi pesanan dari customers. Data yang tersimpan pada tabel ini berupa data waktu customers orders, status orders, hingga waktu pesanan customers tiba (diterima).

-   payments\_dataset :

tabel ini menyimpan informasi mengenai tipe – tipe pembayaran yang digunakan oleh customers saat melakukan pembelanjaan.

-   products\_dataset :

tabel ini menyimpan informasi mengenai detail product yang dibeli oleh customers. Ini berupa nama product, category product, dll.

-   reviews\_dataset :

tabel ini menyimpan informasi mengenai review yang diberikan oleh customer setelah mereka membeli product dari ecommerce.

-   sellers\_dataset :

tabel ini menyimpan detail informasi mengenai seller yang menjual productnya pada ecommerce.

**TASK 2**

Pada Task 2 ini, kami ingin mencari informasi mengenai Annual Customer Activity Growth Analysis. Ini merupakan informasi mengenai perkembangan jumlah customer pada layanan ecommerce setiap tahunnya. Analisis ini membuat beberapa poin yakni :

* Rata – Rata jumlah customers aktif bulanan (Monthly Active User) untuk setiap tahunnya.
* Total Customers baru per tahun.
* Jumlah customers yang melakukan repeat order per tahun.
* Rata – rata frequency order untuk setiap tahun.

Pada task ini juga, didefinisikan sebuah CTE (tabel sementara) yang digunakan sebagai tabel utama dalam melakukan analisis ecommerce, tabel ini akan selalu digunakan pada setiap tasknya. Oleh karena itu penggunaan CTE sangat diperlukan, berikut alasan penggunaan CTE :

1. Kode Mudah Dibaca.
2. Kemudahan Dalam Pengelompokan Data.
3. Penggunaan Kode Berulang – Ulang.
4. Optimasi Kinerja

Berikut ini merupakan tabel CTE (master\_table) yang akan digunakan pada task ini dan task berikutnya :

WITH master\_table AS(

    SELECT

        database\_customers.customer\_id,

        database\_customers.customer\_unique\_id,

        database\_order.order\_id,

        database\_order.order\_status,

        database\_order.order\_purchase\_timestamp AS tanggal\_order,

        database\_order\_item.order\_item\_id AS jumlah\_order,

        database\_order\_item.price,

        database\_order\_item.freight\_value,

        database\_product.product\_id,

        database\_product.product\_category\_name

    FROM

        public.customers\_dataset AS database\_customers

    JOIN

        public.orders\_dataset AS database\_order

    ON

        database\_order.customer\_id = database\_customers.customer\_id

    JOIN

        public.order\_items\_dataset AS database\_order\_item

    ON

        database\_order\_item.order\_id = database\_order.order\_id

    JOIN

        public.products\_dataset AS database\_product

    ON

        database\_product.product\_id = database\_order\_item.product\_id

),

Part 1 :

Part ini akan mencari nilai dari rata – rata customers aktif bulanan untuk setiap tahunnya, informasi ini digunakan untuk melihat seberapa baik dan banyak user yang aktif pada setiap bulannya. Semakin banyak customers aktif maka ini menandakan bahwa Perusahaan memiliki prospek yang baik terhadap pendapatan Perusahaan, dimana Perusahaan memiliki kolam atau pelanggan yang berpeluang melakukan transaksi pada layanan ecommerce kami.

Berikut merupakan query untuk mencari rata – rata customers aktif bulanan pada setiap tahunnya (MAU) :

monthly\_active\_user AS(

    SELECT

        EXTRACT(YEAR FROM tanggal\_order) AS tahun,

        COUNT(DISTINCT customer\_id) AS monthly\_active\_user

    FROM

        master\_table

    GROUP BY

        1

),

Jumlah Monthly Acitive User merujuk pada kolom customer\_id, kolom ini menyimpan informasi id dari customer tetapi tidak bersifat unik. Maka dari itu digunakan sebuah fungsi DISTINCT untuk mendapatkan customer\_id yang bersifat unik.

**Part 2 :**

Pada part 2, kami ingin mengetahui seberapa banyak user baru yang join pada layanan ecommerce kami, informasi ini akan digunakan untuk mengetahui seberapa besar progress perkembangan Perusahaan dalam memperluas pasar dengan cara mengukur banyaknya customer yang join pada setiap tahunnya. Semakin banyak customer yang melakukan join pada setiap tahunnya, maka semakin baik kondisi perusahaan dalam memperluas atau menjangkau pelanggan – pelanggan dan ini menandakan perusahaan memiliki pertumbuhan yang positif pada setiap tahunnya.

Berikut merupakan query untuk mencari informasi mengenai total customer baru pada setiap tahunnya :

new\_customers AS(

    SELECT

        EXTRACT(YEAR FROM tanggal\_order) AS tahun,

        COUNT(DISTINCT(customer\_unique\_id)) AS new\_customers

    FROM

        master\_table

    GROUP BY

        1

),

Jumlah new customer merujuk pada kolom customer\_unique\_id, kolom ini menyimpan informasi id dari customer yang bersifat unik. Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

**Part 3 :**

Part 3 merupakan part yang akan mencari informasi mengenai total customer yang melakukan repeat order pada setiap tahunnya. Informasi repeat order ini merupakan parameter yang digunakan untuk mengukur seberapa baik layanan dari ecommerce kami terhadap customer – customer. Semakin banyak total customer yang melakukan repeat order pertahunnya, maka ini menandakan semakin banyak pula customer yang puas dan senang terhadap layanan dari ecommerce kami. Selain itu, repeat order menandakan banyak customer tersebut percaya atau yakin terhadap Perusahaan kami yang dibuktikan bahwa customer tersebut melakukan beberapa order tanpa merasa cemas barang yang mereka pesan hilang atau tidak sesuai.

Berikut merupakan query untuk mencari jumlah customer yang melakukan repeat order per tahunnya pada layanan ecommerce kami :

repeat\_order AS(

    SELECT

        EXTRACT(YEAR FROM tanggal\_order) AS tahun,

        COUNT(DISTINCT customer\_id) - COUNT(DISTINCT customer\_unique\_id) AS total\_repeat\_order

    FROM

        master\_table

    GROUP BY

        1

),

Total repeat order didapatkan dari perhitungan antara kolom customer\_id (id customer) dikurangi dengan dengan kolom customer\_unique\_id.

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Part 4 :

Pada part ini, akan dicari sebuah informasi mengenai rata – rata frequency order pertahunnya. Informasi ini digunakan untuk mengetahui apakah terjadi penurunan rata – rata penjualan pada setiap tahunnya.

Berikut merupakan query untuk mencari informasi mengenai frequency order pertahunnya :

frequency\_order AS(

    SELECT

        EXTRACT(YEAR FROM tanggal\_order) AS tahun,

        ROUND(AVG(jumlah\_order), 3) AS average\_freq\_order

    FROM

        master\_table

    GROUP BY

        1

)

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

**Part 5 :**

Pada part ini, kami menggabungkan setiap part dari part 1 hingga part 4, ini dilakukan untuk mendapatkan informasi berupa satu buah tabel akhir yang menyimpan informasi dari part 1 hingga part 4. Di part ini kami akan mendapatkan informasi atau tabel utama berupa Annual Customer Activity Growth Analysis..

Berikut query untuk mendapatkan informasi mengenai Annual Customer Activity Growth Analysis :

SELECT

    mau.tahun,

    mau.monthly\_active\_user,

    nc.new\_customers,

    ro.total\_repeat\_order,

    fo.average\_freq\_order

FROM

    monthly\_active\_user mau

JOIN

    new\_customers nc ON mau.tahun = nc.tahun

JOIN

    repeat\_order ro ON nc.tahun = ro.tahun

JOIN

    frequency\_order fo ON ro.tahun = fo.tahun

Pada part ini, untuk medapatkan tabel utama Annual Customer Activity Growth Analysis. Maka harus dilakukan join dari part 1 hingga part 4.

**TASK 3**

**Part 1 :**

Pada bagian ini kita akan menggunakan tabel utama berupa Tabel CTE (master\_table) yang sama dengan task sebelumnya. Tabel CTE ini digunakan sebagai rujukan awal dalam analisis **Annual Product Category Quality Analysis.**

Untuk part pertama akan dilakukan analisis berupa total revenue yang didapatkan oleh Perusahaan setiap tahunnya. Revenue merujuk pada harga barang (price) + harga ongkir barang (freight\_value).

Berikut merupakan query total revenue perusahaan per tahun :

revenue\_table AS(

    SELECT

        EXTRACT(YEAR FROM tanggal\_order) AS tahun,

        ROUND(SUM(CAST((price + freight\_value) AS numeric)), 2) AS total\_revenue\_year

    FROM

        master\_table

    WHERE

        order\_status = 'delivered'

    GROUP BY

        1

),

Query yang dibuat menggunakan teknik CTE, dimana teknik ini digunakan untuk memudahkan proses penggunaan berulang – ulang sebuah query. Selain itu menggunakan CTE ini akan menghemat ruang penyimpanan database dari Perusahaan karena tabel CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Pada kasus ini terdapat klausa berupa order\_status = delivered, karena kami beropini bahwa revenue suatu Perusahaan didapatkan ketika suatu produk tersebut berhasil terjual sepenuhnya tanpa ada return ataupun canceled, sebaliknya apabila terdapat canceled maka ini merupakan sebuah kerugian bagi perusahaan.

**Part 2 :**

Pada part ini, kami akan menghitung berapa banyak total barang yang mengalami canceled order pada setiap tahunnya. Perhitungan ini digunakan untuk mengetahui berapa banyak barang yang menyebabkan kerugian bagi perusahaan dan dapat menjadi acuan estimasi banyaknya customers yang mengalami experience yang kurang baik ketika melakukan pembelanjaan pada Perusahaan kami.

Berikut merupakan query untuk menghitung total barang yang mengalami canceled order per tahun :

canceled\_table AS(

    SELECT

        EXTRACT(YEAR FROM tanggal\_order) AS tahun,

        SUM(jumlah\_order) AS total\_canceled\_year

    FROM

        master\_table

    WHERE

        order\_status = 'canceled'

    GROUP BY

        1

    ORDER BY

        1 DESC

),

Total canceled didapatkan dari kolom jumlah\_order, dimana kolom ini berisi total produk yang dibeli oleh customer dalam 1 kali transaksi. Pada kasus ini terdapat klausa berupa order\_status = canceled, klausa ini digunakan karena klausa ini menunjukkan bahwa product tersebut mengalami canceled order oleh customer.

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

**Part 3 :**

Pada part ini, dilakukan sebuah analisis terhadap top category product yang memberikan revenue paling tinggi untuk perusahaan. Analisis ini dilakukan guna mendapatkan informasi tentang category product apa yang sangat banyak memberikan pendapatan perusahaan, category product tersebut akan dijadikan sebuah product unggulan bagi Perusahaan kedepannya dan product tersebut akan dijadikan sebagai product senjata utama untuk mendapatkan pendapatan yang lebih tinggi pada tahun – tahun berikutnya.

Berikut merupakan query untuk menemukan top product category yang memberikan revenue tertinggi bagi Perusahaan setiap tahunnya :

product\_top\_revenue AS(

    SELECT

        EXTRACT(YEAR FROM tanggal\_order) AS tahun,

        product\_category\_name AS product\_category\_revenue,

        ROUND(SUM(CAST((price + freight\_value) AS numeric)), 2) AS total\_revenue\_product,

        RANK() OVER(PARTITION BY

                    EXTRACT(YEAR FROM tanggal\_order)

                    ORDER BY ROUND(SUM(CAST((price + freight\_value) AS numeric)), 2) DESC) AS top\_rank

    FROM

        master\_table

    WHERE

        order\_status = 'delivered' AND product\_category\_name IS NOT NULL

    GROUP BY

        1, 2

    ORDER BY

        1, 4 ASC

),

Total revenue yang digunakan pada analisis ini merujuk pada harga product (price) + harga ongkir product (freight\_value). Setelah didapatkan total revenue setiap product category, maka akan diambil product category yang memiliki urutan teratas (rank 1) pada setiap tahunnya. Pada kasus ini terdapat klausa berupa order\_status = delivered, karena kami beropini bahwa revenue didapatkan dari hasil keberhasilan transaksi customer atau hingga customer mendapatkan barang yang mereka pesan. Selain itu, terdapat klausa IS NOT NULL ini digunakan untuk mengembalikan product category name yang terdapat value (nama) pada setiap barisnya atau kami menolak untuk mendapatkan revenue Perusahaan tetapi tidak diketahui product category namenya.

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

**Part 4 :**

Pada part ini, kami melakukan analisis mengenai top product category name yang paling banyak mengalami canceled order pada setiap tahunnya. Analisis ini dilakukan untuk mengetahui product apa saja yang menjadi penyebab utama perusahaan mengalami kerugian pada setiap tahunnya. Informasi category product yang didapatkan akan dijadikan sebagai bahan evaluasi untuk tahun – tahun berikutnya agar product tersebut tidak selalu mengalami canceled order paling banyak.

Berikut merupakan query untuk mencari top category product name yang mengalami canceled paling banyak setiap tahunnya :

product\_top\_canceled AS(

    SELECT

        EXTRACT(YEAR FROM tanggal\_order) AS tahun,

        product\_category\_name AS product\_category\_canceled,

        SUM(jumlah\_order) AS total\_canceled\_product,

        RANK() OVER(PARTITION BY EXTRACT(YEAR FROM tanggal\_order)

                    ORDER BY SUM(jumlah\_order) DESC) AS top\_rank

    FROM

        master\_table

    WHERE

        order\_status = 'canceled'

    GROUP BY

        1, 2

    ORDER BY

        1, 4

)

Total canceled product didapatkan dari hasil perhitungan kolom jumlah order, kolom ini berisikan total produk yang dibeli oleh customer pada saat 1 kali transaksi. Setelah didapatkan total canceled setiap product category, maka akan diambil product category name yang memiliki urutan teratas (rank 1) pada setiap tahunnya. Pada part ini menggunakan klausa order\_status = canceled karena klausa ini menunjukkan bahwa product tersebut mengalami canceled order oleh customer

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

**Part 5 :**

Pada part ini, kami menggabungkan setiap part dari part 1 hingga part 4, ini dilakukan untuk mendapatkan informasi berupa satu buah tabel akhir yang menyimpan informasi dari part 1 hingga part 4. Di part ini kami akan mendapatkan informasi atau tabel utama berupa Annual Product Category Quality Analysis.

Berikut ini merupakan query untuk mendapatkan tabel Annual Category Product Quality Analysis pertahunnya :

SELECT

    rt.tahun,

    ptr.product\_category\_revenue,

    ptr.total\_revenue\_product,

    rt.total\_revenue\_year,

    ptc.product\_category\_canceled,

    ptc.total\_canceled\_product,

    ct.total\_canceled\_year

FROM

    revenue\_table rt

JOIN

    canceled\_table ct ON rt.tahun = ct.tahun

JOIN

    product\_top\_revenue ptr ON ct.tahun = ptr.tahun

JOIN

    product\_top\_canceled ptc ON ptr.tahun = ptc.tahun

WHERE

    ptr.top\_rank = 1 AND ptc.top\_rank = 1

ORDER BY

    1

Pada part ini, untuk medapatkan tabel utama Annual Category Product Quality Analysis. Maka harus dilakukan join dari part 1 hingga part 4.

**TASK 4**

**Part 1 :**

Pada part ini, kami melakukan sebuah analisis tentang type pembayaran yang sering digunakan oleh customer pada saat melakukan transaksi. Analisis ini digunakan untuk mendapatkan informasi tentang payment type usage, informasi ini dapat digunakan untuk memberikan peningkatan penjualan pada perusahaan dimana kami dapat memberikan promo kepada customer apabila melakukan transaksi dengan metode tertentu.

Berikut ini merupakan query untuk mendapatkan informasi mengenai Analysis of Annual Payment Type Usage per tahun :

WITH payments\_fav AS(

    SELECT

        EXTRACT(YEAR FROM od.order\_purchase\_timestamp) AS tahun,

        pd.payment\_type,

        COUNT(2) AS total\_used

    FROM

        payments\_dataset pd

    JOIN

        orders\_dataset od ON pd.order\_id = od.order\_id

    GROUP BY

        1, 2

    ORDER BY

        1, 3 DESC

)

Query diatas merupakan query untuk pembuatan tabel sementara (CTE), tabel ini digunakan untuk rujukan analisis payment type usage. Pada query diatas dilakukan join antara dua tabel yakni tabel orders\_dataset untuk mengambil nilai pada kolom order\_purchase\_timestamp (informasi tanggal transaksi) dan tabel payments\_dataset untuk mengambil nilai payment\_type (tipe pembayaran yang digunakan customer).

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

**Part 2 :**

Pada part 2, kami melakukan query untuk mendapatkan informasi berupa sebuah tabel pivot yang berisi nilai tipe pembayaran yang digunakan customer pada setiap tahunnya.

Berikut merupakan query untuk mendapatkan tabel Annual Payment Type Usage pertahun :

SELECT

    payment\_type,

    SUM(CASE WHEN tahun = 2016 THEN total\_used ELSE 0 END) AS tahun\_2016,

    SUM(CASE WHEN tahun = 2017 THEN total\_used ELSE 0 END) AS tahun\_2017,

    SUM(CASE WHEN tahun = 2018 THEN total\_used ELSE 0 END) AS tahun\_2018

FROM

    payments\_fav

GROUP BY

    1