JURNAL RISET SISTEM INFORMASI
Halaman Jurnal: https://journal.smartpublisher.id/index.php/jissi
Halaman UTAMA Jurnal : https://journal.smartpublisher.id

DOI: https://doi.org/10.69714/8dwt6j74

PERANCANGAN *DATA WAREHOUSE E-COMMERCE* MENGGUNAKAN POSTGRESQL PADA *PUBLIC DATASET* OLIST

Arron Mosses Jhon Hadi[a], Ayu Elisya Natama Sianturi[b], Andri Wijaya[c*]
[a] Fakultas Sains dan Teknologi / Sistem Informasi; arronmosses31@gmail.com, Universitas Katolik Musi Charitas; Jalan Bangau No. 60 Palembang, Sumatera Selatan
[b] Fakultas Sains dan Teknologi / Sistem Informasi; ayuelisyans@gmail.com, Universitas Katolik Musi Charitas; Jalan Bangau No. 60 Palembang, Sumatera Selatan
[c] Fakultas Sains dan Teknologi / Sistem Informasi; andri_wijaya@ukmc.ac.id, Universitas Katolik Musi Charitas; Jalan Bangau No. 60 Palembang, Sumatera Selatan
[*] Penulis Korespondensi: Andri Wijaya

ABSTRACT

This study designs a PostgreSQL-based data warehouse using the Olist public dataset to address fragmented and unstructured e-commerce transactional data. The research process includes ETL (Extract, Transform, Load), data cleaning and standardization, table consolidation, and the development of a star schema consisting of a sales fact table and multiple dimension tables. OLAP analysis reveals key patterns such as annual sales trends, top product categories, seller performance, preferred payment methods, and customer geographic distribution. The results demonstrate that the data warehouse improves analytical efficiency and provides strategic insights to support business intelligence in the e-commerce environment
.

Keywords: *data warehouse*; *e-commerce*; ETL; *star schema*; OLAP

Abstrak

Penelitian ini merancang *data warehouse* berbasis PostgreSQL menggunakan *public dataset* Olist untuk mengatasi permasalahan data transaksi *e-commerce* yang tersebar dan tidak terstruktur. Proses penelitian mencakup ETL (*Extract, Transform, Load*), pembersihan dan standardisasi data, penggabungan tabel operasional, serta pembangunan *star schema* yang terdiri dari tabel fakta penjualan dan beberapa tabel dimensi. Hasil OLAP menunjukkan pola penting seperti tren penjualan tahunan, kategori produk terlaris, performa seller, preferensi metode pembayaran, dan persebaran pelanggan berdasarkan lokasi. Perancangan ini membuktikan bahwa *data warehouse* mampu meningkatkan efisiensi analisis dan menghasilkan insight strategis bagi kebutuhan *business intelligence* pada lingkungan *e-commerce*.

Kata Kunci: *data warehouse*; *e-commerce*; ETL; *star schema*; OLAP

1. PENDAHULUAN

Pertumbuhan pesat platform *e-commerce* dalam beberapa tahun terakhir telah menghasilkan volume data transaksi dan interaksi pengguna yang besar, cepat, dan heterogen. Data dari aktivitas seperti pencarian produk, pemesanan, pembayaran, pengiriman, dan ulasan menyimpan informasi bernilai untuk memahami perilaku pelanggan, memprediksi permintaan, serta mengevaluasi performa bisnis. Namun, data operasional pada basis OLTP umumnya terfragmentasi dan tidak dioptimalkan untuk kebutuhan analitik jangka panjang; akibatnya analisis historis dan *query* multidimensi sering kali lambat, tidak efisien, dan rentan terhadap inkonsistensi format atau duplikasi data [1][2].

Untuk mengatasi kendala tersebut, dibutuhkan arsitektur *data warehouse* yang menyediakan repositori terpusat yang mengintegrasikan berbagai sumber operasional, menyimpan data historis, dan dioptimalkan untuk kebutuhan analitik hingga *business intelligence* (BI). Implementasi *data warehouse* modern juga

Naskah Masuk 8 Desember 2025; Revisi 10 Desember 2025; Diterima 14 Desember 2025; Tersedia 18 Desember 2025

berinteraksi dengan konsep *data lake/lakehouse* ketika menghadapi volume dan variasi data besar, sehingga strategi integrasi (ETL/ELT/ECLT) menjadi sangat penting untuk menjaga performa dan kualitas data bagi keperluan analitik [3][4].

Dalam praktik perancangan *data warehouse*, pemodelan dimensional termasuk *star schema* masih menjadi pendekatan yang efisien untuk memetakan fakta bisnis (mis. transaksi penjualan) dan dimensi analitik (mis. waktu, pelanggan, produk, lokasi). Skema ini menyederhanakan *query* OLAP dan memudahkan agregasi pada berbagai tingkat granularitas, sehingga lazim dipakai pada studi kasus retail dan *e-commerce*. Selain itu, penelitian terkini juga menekankan perlunya optimasi *pipeline* ETL dan teknik penyimpanan/partisi untuk menjaga kinerja *query* pada sistem *data warehouse* (termasuk ketika diimplementasikan di RDBMS seperti PostgreSQL atau di lingkungan *cloud*) [5][6].

Meskipun berbagai penelitian sebelumnya telah membahas penerapan *data warehouse* dan teknik ETL pada berbagai domain, sebagian besar studi tersebut berfokus pada konteks institusi tertentu atau menggunakan data internal dengan skala terbatas. Selain itu, implementasi *data warehouse* pada domain *e-commerce* sering kali dibahas secara konseptual tanpa mengeksplorasi secara mendalam proses integrasi data yang kompleks dan heterogen dari dataset publik berskala besar. Penggunaan PostgreSQL sebagai *platform data warehouse* juga masih relatif jarang dikaji secara spesifik pada dataset *e-commerce* terbuka, khususnya dalam konteks pemodelan dimensional dan analisis OLAP berbasis transaksi riil.

Oleh karena itu, penelitian ini mengisi celah (*research gap*) dengan merancang dan mengimplementasikan *data warehouse e-commerce* berbasis PostgreSQL menggunakan *Brazilian E-Commerce Public Dataset* (Olist) yang bersifat publik dan kompleks. Kontribusi utama penelitian ini terletak pada penyajian proses ETL yang sistematis, penerapan *star schema* pada data transaksi *e-commerce*, serta demonstrasi kemampuan PostgreSQL dalam mendukung analisis OLAP untuk menghasilkan insight bisnis yang relevan. Dengan demikian, penelitian ini tidak hanya memberikan kontribusi praktis bagi pengembangan sistem *business intelligence*, tetapi juga memperkaya kajian akademik terkait implementasi data warehouse pada domain *e-commerce* berbasis data publik.

2. TINJAUAN PUSTAKA

2.1. *Data Warehouse*

Data warehouse adalah sistem repositori terpusat yang dirancang khusus untuk mendukung analisis historis dan pengambilan keputusan. Berbeda dengan basis data operasional, *data warehouse* mengintegrasikan data dari berbagai sumber, menyajikan data yang sudah dibersihkan dan terstruktur sehingga memudahkan analitik dan pelaporan [7]. Studi-studi penerapan *data warehouse* pada konteks institusi pendidikan, pemerintahan, maupun bisnis menunjukkan manfaat yang konsisten dalam meningkatkan kualitas laporan, konsistensi data, dan kecepatan analisis operasional [8][9].

Dalam konteks *e-commerce*, kebutuhan integrasi data menjadi sangat penting karena transaksi dan interaksi pengguna tersebar di banyak tabel dan sumber (*orders, payments, items, reviews, seller info*). Tanpa repositori analitik, proses *query* agregasi dan pelaporan menjadi lambat dan rawan inkonsistensi. Oleh karena itu, *data warehouse* menyediakan fondasi untuk transformasi data operasional menjadi informasi strategis yang mendukung pengambilan keputusan berbasis data (*data-driven decision making*) [10][11] .

Dalam penelitian ini, konsep *data warehouse* tersebut diterapkan untuk mengintegrasikan data transaksi *e-commerce* Olist yang awalnya tersebar pada berbagai tabel operasional. Pendekatan ini memungkinkan transformasi data transaksi *e-commerce* menjadi struktur analitik yang mendukung analisis historis dan pengambilan keputusan strategis secara lebih efisien.

2.2. Skema Dimensional *Star Schema*

Desain skema dimensional merupakan pendekatan praktis untuk memodelkan *data warehouse*. *Star schema* menempatkan satu tabel fakta di pusat (mengandung metrik kuantitatif transaksi) dan dikelilingi oleh beberapa tabel dimensi yang berisi atribut deskriptif (misalnya waktu, produk, pelanggan, penjual, lokasi). Keunggulan *star schema* adalah sederhana, mudah dipahami, dan dioptimalkan untuk operasi OLAP seperti agregasi dan *drill-down*, sehingga sering dipilih dalam studi dan implementasi *data warehouse* skala menengah hingga besar [8]. Beberapa penelitian yang menerapkan *star schema* menunjukkan peningkatan performa *query* analitik serta kemudahan pembuatan laporan bisnis [9][12].

Berdasarkan keunggulan tersebut, penelitian ini mengadopsi *star schema* sebagai model utama perancangan *data warehouse* karena fokus analisis hanya pada satu proses bisnis inti, yaitu transaksi penjualan *e-commerce*. Pemilihan ini diharapkan mampu menyederhanakan struktur data serta meningkatkan performa *query* OLAP pada PostgreSQL.

2.3. Proses ETL (*Extract*, *Transform*, *Load*)

ETL adalah proses inti yang menjamin kualitas data dalam *data warehouse*, meliputi tahap pengambilan data dari sumber (*Extract*), pembersihan dan pemetaan atribut (*Transform*), lalu pemuatan ke tabel fakta/dimensi (*Load*). Beberapa penelitian menekankan bahwa desain ETL yang baik secara langsung meningkatkan akurasi hasil analitik dan mengurangi waktu eksekusi *pipeline*, termasuk *metadata management*, *cleansing*, dan transformasi bertingkat. Implementasi ETL dapat menggunakan *tool open source* atau skrip *custom*; pilihan ini bergantung pada volume data, frekuensi pembaruan, dan kompleksitas transformasi yang diperlukan [7][10][11].

Dalam studi kasus yang memanfaatkan *dataset e-commerce* (seperti Olist), tahap ETL biasanya meliputi normalisasi data alamat/lokasi, *mapping* status pesanan, konsolidasi metode pembayaran, dan pembuatan *surrogate key* untuk fakta/dimensi. Langkah-langkah tersebut penting agar analisis multidimensi berjalan konsisten [10][12].

Prinsip-prinsip ETL tersebut menjadi dasar dalam penelitian ini, khususnya pada tahap pembersihan data, standarisasi atribut, dan penggabungan tabel operasional Olist agar data yang dimuat ke dalam *data warehouse* memiliki kualitas dan konsistensi yang memadai untuk analisis multidimensi.

2.4. OLAP (*Online Analytical Processing*)

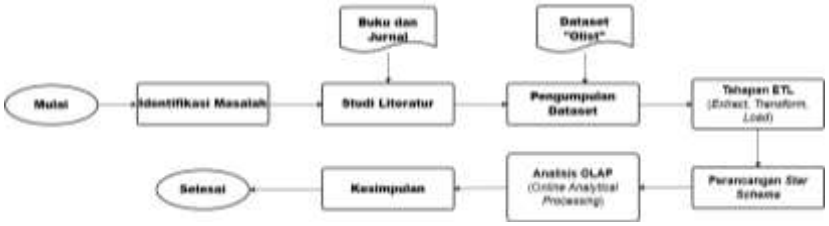
OLAP menyediakan operasi analitik multidimensi seperti *slice, dice, drill-down/roll-up,* dan *pivot* yang digunakan untuk mengeksplorasi data historis secara fleksibel [13]. Implementasi OLAP pada RDBMS seperti PostgreSQL umumnya memanfaatkan fungsi agregasi, *window functions,* serta *GROUPING SETS (ROLLUP/CUBE)* untuk mensimulasikan kubus analitik tanpa memerlukan infrastruktur OLAP khusus. Penelitian menunjukkan bahwa kombinasi *star schema* dan optimasi *query* pada RDBMS meningkatkan performa analitik dan memenuhi kebutuhan BI organisasi skala menengah, terutama pada lingkungan yang memanfaatkan *indexing*, partisi, dan desain dimensional yang tepat [12].

Dalam penelitian ini, operasi OLAP diimplementasikan langsung pada PostgreSQL dengan memanfaatkan struktur *star schema*, sehingga analisis multidimensi dapat dilakukan tanpa memerlukan perangkat OLAP khusus.

3. METODOLOGI PENELITIAN

3.1. Kerangka Penelitian

Kerangka penelitian ini disusun sebagai panduan sistematis dalam perancangan *data warehouse* untuk platform *e-commerce* menggunakan PostgreSQL pada *public dataset* Olist. Penelitian dimulai dengan proses identifikasi masalah untuk menentukan kebutuhan analisis dan tujuan pengembangan *data warehouse*. Selanjutnya dilakukan studi literatur guna memperoleh landasan teori terkait konsep *data warehouse*, ETL, *star schema*, serta analisis OLAP. Setelah itu, *dataset* Olist dikumpulkan sebagai sumber utama yang akan diolah. Tahapan ETL kemudian diterapkan untuk melakukan ekstraksi, pembersihan, transformasi, dan pemuatan data ke dalam sistem. Hasil ETL digunakan dalam perancangan star schema yang menjadi struktur utama penyimpanan data multidimensional. Tahap berikutnya adalah melakukan analisis OLAP untuk menghasilkan informasi yang relevan bagi kebutuhan bisnis. Setelah itu penyusunan kesimpulan berdasarkan hasil perancangan dan analisis yang telah dilakukan.



Gambar 3.1 Kerangka Penelitian

3.2. Sumber Data Penelitian

Data penelitian berasal dari *Brazilian E-Commerce Public Dataset by Olist*, sebuah *dataset* transaksi *e-commerce* yang dirilis secara publik pada *platform Kaggle* dan berisi lebih dari seratus ribu pesanan pada periode 2016 sampai 2018. *Dataset* ini terdiri dari berbagai tabel terpisah seperti *orders*, *order_items*, *order_payments*, *order_reviews*, *products*, *customers*, *sellers*, serta tabel pendukung lain seperti *geolocation*, *leads_closed*, dan *leads_qualified*. Kompleksitas data tersebut menjadikannya sangat ideal sebagai bahan penelitian pembangunan *data warehouse*, karena mencerminkan kondisi nyata bisnis *e-commerce*. Seluruh tabel awalnya berada dalam format SQLite dan kemudian diekspor menjadi CSV sebelum diproses lebih lanjut dalam PostgreSQL.

3.3. Proses ETL (*Extract*, *Transform*, *Load*)

3.3.1. Tahap *Extract*

Tahap *extract* merupakan proses pemindahan data dari sumber awal ke area *staging* tanpa melakukan modifikasi nilai. Seluruh tabel pada database SQLite dibuka dan diekspor satu per satu menjadi file CSV. *File* tersebut kemudian dimasukkan ke PostgreSQL menggunakan perintah *COPY* agar strukturnya tetap sama dengan data asli. Tahap ini memastikan bahwa tidak ada informasi yang hilang selama proses pemindahan dan dataset tersedia dalam kondisi lengkap sebelum dilakukan proses pembersihan.

3.3.2. Tahap *Transform*

Tahap *transform* merupakan bagian paling krusial karena seluruh kualitas data diperbaiki pada fase ini. Banyak kolom dalam *dataset* Olist yang tidak sesuai tipe data yang seharusnya, sehingga dilakukan serangkaian proses pembersihan dan penyesuaian.

Tahap pertama adalah *data cleaning*, yang mencakup penghapusan nilai yang tidak valid, baris yang berisi header ganda, serta string kosong yang harus digantikan dengan *NULL*. Beberapa kolom numerik seperti harga, kuantitas, dan *installment* juga ditemukan berisi karakter non-numerik, sehingga nilai-nilai tersebut dibersihkan terlebih dahulu sebelum dilakukan konversi. Pada kolom tanggal, seluruh nilai yang tidak memenuhi standar format juga dihapus dan sisanya dikonversi ke tipe *timestamp* PostgreSQL.

Tahap berikutnya adalah penyesuaian tipe data agar seluruh kolom sesuai dengan perannya. Kolom harga diubah menjadi tipe *numeric*, jumlah unit menjadi *integer*, dan koordinat geografi menjadi *numeric*. Setelah itu, dilakukan standarisasi nilai *missing*, di mana berbagai bentuk nilai kosong seperti “unknown”, “*none*”, atau *string* kosong diseragamkan menjadi *NULL* agar tidak mengganggu proses analitis.

Setelah memastikan bahwa seluruh tabel siap digunakan, peneliti melakukan penggabungan beberapa tabel operasional menjadi satu tabel baru bernama *fact_sales_source*. Tabel ini berfungsi sebagai sumber utama untuk pembentukan *fact table* karena berisi seluruh informasi transaksi mulai dari pesanan, detail item, pembayaran, hingga ulasan pelanggan. Penggabungan ini memungkinkan alur transaksi dilihat secara utuh dan memudahkan proses transformasi lanjutan.

3.3.3. Tahap *Load*

Tahap *load* dilakukan dengan memindahkan data hasil transformasi ke dalam struktur *data warehouse*. Dua skema dibuat, yaitu *staging* sebagai tempat penyimpanan sementara dan *data warehouse* sebagai penyimpanan permanen untuk tabel fakta dan dimensi. Struktur *data warehouse* dirancang menggunakan *star schema* karena skema ini sederhana, efisien, dan sangat cocok digunakan untuk analisis OLAP.

Pada tahap ini, tabel dimensi seperti *dim_date*, *dim_customer*, *dim_seller*, *dim_product*, dan *dim_payment* dibuat terlebih dahulu. Masing-masing tabel dimensi diberikan *surrogate key* untuk mempercepat proses *join*. Setelah seluruh dimensi selesai diisi, tabel fakta utama yaitu *fact_sales* dibentuk dan diisi dengan data dari *fact_sales_source* yang telah dipetakan terhadap kunci-kunci dimensi. Proses ini menghasilkan struktur *data warehouse* yang stabil, konsisten, dan siap dianalisis lebih lanjut.

3.4. Proses OLAP (*Online Analytical Processing*)

Setelah *data warehouse* berhasil dibangun, analisis dilakukan menggunakan teknik OLAP (*Online Analytical Processing*). Teknik ini memungkinkan peneliti melihat data dari berbagai perspektif seperti waktu, kategori produk, lokasi pelanggan, performa *seller*, atau metode pembayaran. Beberapa teknik

OLAP seperti *roll-up*, *drill-down*, *slice*, *dice*, dan *pivot* digunakan untuk menemukan pola, menghitung agregasi, serta memahami hubungan antarvariabel. Hasil analisis ini kemudian divisualisasikan ke dalam bentuk *dashboard* agar lebih mudah dipahami oleh pengguna maupun manajemen.

3.5. Justifikasi Pemilihan *Platform* dan Skema Dimensional

Pemilihan PostgreSQL dalam penelitian ini didasarkan pada kemampuan platform tersebut untuk menangani beban analitis yang tinggi melalui dukungan *window functions, parallel query, indexing* yang fleksibel, serta stabilitas yang cocok untuk pengolahan data berskala menengah seperti dataset Olist. PostgreSQL juga bersifat *open-source*, mudah diintegrasikan dalam *pipeline* ETL, serta memiliki performa yang baik untuk *workload* yang dominan membaca data, sehingga sesuai dengan karakter *data warehouse*. Sementara itu*, star schema* dipilih karena struktur ini sederhana, mudah dipahami, dan memberikan performa terbaik untuk *query* agregasi yang umum digunakan pada analisis *e-commerce*. Dibandingkan alternatif seperti *snowflake schema* atau *galaxy schema*, *star schema* lebih efisien karena fokus penelitian hanya pada satu proses bisnis utama, yaitu transaksi penjualan. Pemilihan *fact table* dan *dimension table* dilakukan dengan mempertimbangkan granularitas transaksi (*order item level*), kebutuhan analitis, dan stabilitas atribut. Data operasional yang awalnya tersebar kemudian direstrukturisasi menjadi satu tabel fakta dan beberapa dimensi agar analisis multidimensi dapat berjalan lebih cepat, konsisten, dan terorganisasi.

Tabel 3.1 Strukur Data Sebelum Transformasi

Nama Tabel	Jumlah Kolom	Deskripsi Singkat
orders	9	Informasi utama pesanan pada level order.
order_items	7	Detail item yang dibeli dalam setiap pesanan.
order_payments	5	Informasi metode dan nilai pembayaran.
order_reviews	5	Skor penilaian dan komentar pelanggan.
customers	5	Identitas dan lokasi pelanggan.
sellers	4	Informasi penjual dan lokasi.
products	8	Atribut dan kategori produk.
geolocation	**5**	**Data lokasi berupa koordinat kota dan negara bagian.**

Tabel 3.2 Struktur Data Sesudah Transformasi

Nama Tabel	Jenis	Fungsi	Contoh Atribut
fact_sales	Fakta	Menyimpan metrik transaksi pada level order item.	*price, freight_value, payment_value, quantity*
dim_date	Dimensi	Analisis waktu dan periode penjualan.	*date, month, year, quarter*
dim_customer	Dimensi	Deskripsi dan lokasi pelanggan.	*customer_unique_id, city, state*
dim_seller	Dimensi	Informasi penjual.	*seller_id, seller_city, seller_state*
dim_product	Dimensi	Atribut produk dan kategorinya.	*product_id, product_category_name*
dim_payment	Dimensi	Preferensi metode pembayaran.	*payment_type, payment_installments*
dim_review	Dimensi	Informasi ulasan pelanggan.	*review_score, review_creation_date*
dim_geolocation	Dimensi	Lokasi geografis rinci.	*geolocation_city, geolocation_state*

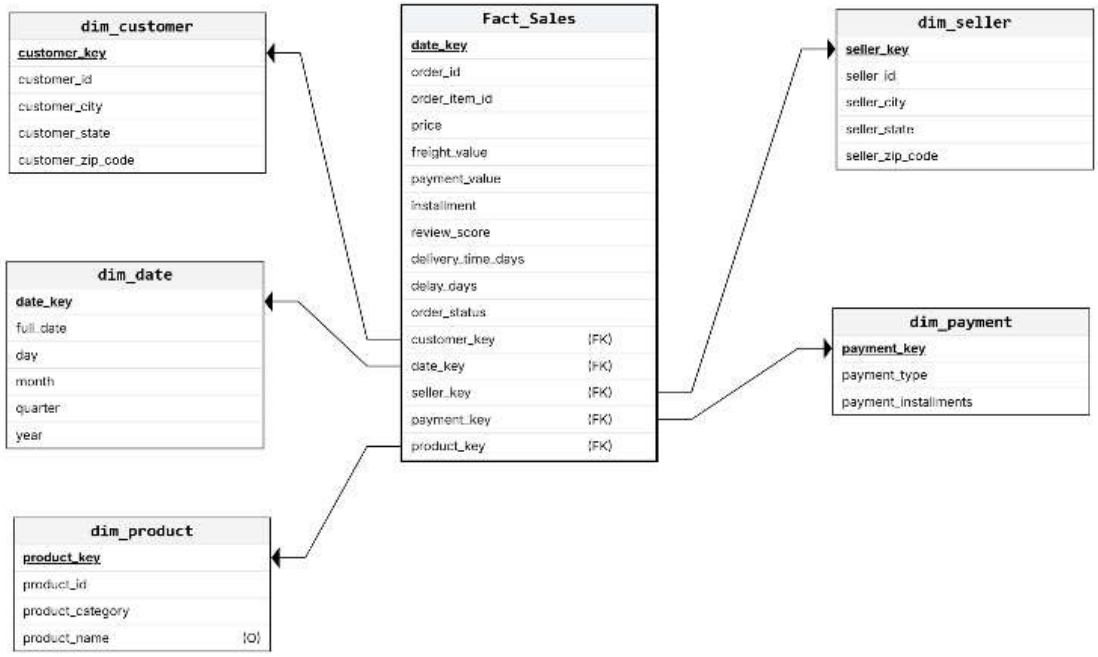
4. HASIL DAN PEMBAHASAN

4.1 Integrasi Data Operasional ke Dalam *Star Schema*

Dataset Olist awalnya tersimpan dalam beberapa tabel operasional seperti *orders, order_items, payments, reviews*, dan *customers*. Setiap tabel memiliki fungsi tertentu dan tidak dirancang untuk analisis multidimensi. Melalui model *data warehouse*, seluruh data tersebut direstrukturisasi menjadi tabel fakta dan tabel dimensi yang saling terhubung melalui *surrogate key*.

Tabel fakta utama adalah *fact_sales*, yang berisi metrik dasar penjualan seperti harga, nilai pembayaran, dan biaya pengiriman. Sementara itu, informasi deskriptif yang tidak berubah dalam transaksi dikelompokkan ke dalam tabel dimensi seperti *dim_customer*, *dim_product*, *dim_seller*, *dim_payment*, *dim_review*, dan *dim_geolocation*.

Dengan pola *Star Schema*, relasi antar data menjadi lebih sederhana dibandingkan struktur operasional aslinya. Hal ini mempermudah pengguna melakukan analisis berdasarkan berbagai perspektif bisnis, seperti waktu pembelian, lokasi pelanggan, kategori produk, hingga performa masing-masing *seller*.



Gambar 4.1 Perancangan *star schema*

4.2 Proses ETL (*Extract*, *Transform*, *Load*)

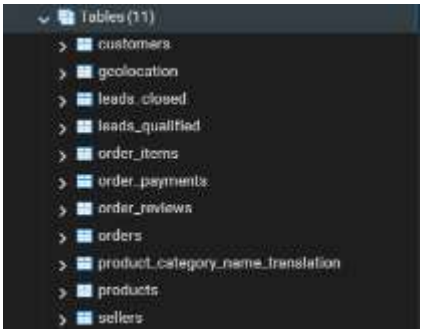
Proses ETL merupakan komponen penting dalam pembentukan *data warehouse* karena menentukan kualitas dan integritas data yang masuk ke dalam sistem analitis. Proses ETL pada penelitian ini dilakukan dalam tiga tahapan.

4.2.1. Tahap *Extract*

Tahap ini dilakukan dengan mengekstraksi seluruh tabel dari file *olist.sqlite* ke format CSV dengan menggunakan DB Browser for SQLite dan kemudian mengimpor seluruh file tersebut ke PostgreSQL melalui *schema staging*. Pada tahap ini tidak ada perubahan terhadap struktur data karena fokus utama adalah memindahkan informasi dari sistem operasional ke area pemrosesan awal.

Gambar 4.2 *File* sumber data

Gambar 4.3 *Dataset* di-*extract* ke dalam format csv



Gambar 4.4 *Dataset* telah di-*export* ke PostgreSQL

4.2.2. Tahap *Transform*

Tahap transformasi mencakup serangkaian proses untuk membersihkan, menata, dan mempersiapkan data agar siap dimuat ke dalam *data warehouse*. Adapun langkah-langkah transformasi yang dilakukan adalah sebagai berikut:

1. *Data Cleaning*

Pada tahap ini dilakukan penghapusan data duplikat, perbaikan format tanggal, penyesuaian tipe data, serta penanganan nilai kosong. Proses ini memastikan bahwa data yang masuk ke *data warehouse* memiliki kualitas yang baik dan konsisten.

	customer_id character varying	customer_unique_id character varying	customer_zip_code_prefix character varying	customer_city character varying	customer_state character varying
1	customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state

Gambar 4.5 Duplikasi nama *column* sebagai *field* di semua tabel

```sql
DELETE FROM staging.customers
WHERE customer_id = 'customer_id'
   OR customer_unique_id = 'customer_unique_id'
   OR customer_zip_code_prefix = 'customer_zip_code_prefix'
   OR customer_city = 'customer_city'
   OR customer_state = 'customer_state';
```

Gambar 4.6 *Query* untuk menghapus duplikasi

```sql
UPDATE staging.orders
SET order_id = NULLIF(order_id, ''),
    customer_id = NULLIF(customer_id, ''),
    order_status = NULLIF(order_status, ''),
    order_purchase_timestamp = NULLIF(order_purchase_timestamp, ''),
    order_approved_at = NULLIF(order_approved_at, ''),
    order_delivered_carrier_date = NULLIF(order_delivered_carrier_date, ''),
    order_delivered_customer_date = NULLIF(order_delivered_customer_date, ''),
    order_estimated_delivery_date = NULLIF(order_estimated_delivery_date, '');
```

Gambar 4.7 *Query* penanganan nilai kosong

```sql
ALTER TABLE staging.order_items
ALTER COLUMN shipping_limit_date TYPE timestamp
USING shipping_limit_date::timestamp;
```

Gambar 4.8 *Query* perbaikan format tanggal

```sql
ALTER TABLE staging.order_items
ALTER COLUMN order_item_id TYPE integer USING order_item_id::integer;

ALTER TABLE staging.order_items
ALTER COLUMN price TYPE numeric USING price::numeric;

ALTER TABLE staging.order_items
ALTER COLUMN freight_value TYPE numeric USING freight_value::numeric;
```

Gambar 4.9 *Query* penyesuaian tipe data

2. Standardisasi

Beberapa atribut pada dataset Olist, khususnya kategori produk dan data lokasi, memiliki variasi penulisan. Oleh karena itu, dilakukan standardisasi untuk menyamakan format penulisan, menggabungkan kategori yang serupa, dan memperbaiki data yang tidak valid.

```
UPDATE staging.customers
SET customer_city = lower(trim(customer_city)),
    customer_state = upper(trim(customer_state));
```

Gambar 4.10 Standardisasi atribut teks

```
UPDATE staging.orders
SET order_status = trim(both ' ' from order_status);
```

Gambar 4.11 Standardisasi spasi berlebihan

```
UPDATE staging.orders
SET order_status = lower(order_status);
```

Gambar 4.12 Standardisasi kategori serupa

3. Penggabungan Tabel (*Joining*)

Tabel-tabel operasional seperti *orders*, *order_items*, *order_payments*, *order_reviews*, serta *customers* digabungkan berdasarkan *primary key* dan *foreign key* untuk membentuk satu alur transaksi yang utuh. Hasil penggabungan ini menjadi dasar pembentukan tabel fakta.

4. Pemetaan Data ke Struktur Dimensional

Setelah seluruh data dibersihkan dan distandardisasi, data dipetakan ke dalam tabel dimensi seperti *dim_customer*, *dim_product*, *dim_seller*, *dim_payment*, *dim_review*, *dim_geolocation*, dan *dim_date*. Sementara itu, data transaksi disiapkan untuk dimuat sebagai tabel fakta.

Gambar 4.13 Pemetaan tabel dimensi

Melalui keempat langkah tersebut, data operasional yang semula tersebar berhasil diubah menjadi data terstruktur yang memenuhi standar *data warehouse*.

4.2.3. Tahap *Load*

Tahap *load* merupakan langkah terakhir dalam proses ETL (*Extract, Transform, Load*). Pada tahap ini, seluruh data yang telah melalui proses pembersihan, standarisasi, transformasi, dan pembentukan struktur *star schema* akan dimasukkan ke dalam *schema* sebagai *data warehouse* utama. Tahap ini memastikan bahwa data sudah tersimpan secara permanen, konsisten, dan siap digunakan untuk keperluan analisis maupun *business intelligence*.

	fact_id [PK] integer	date_key integer	customer_key integer	seller_key integer	product_key integer	payment_key integer	order_id character varying (50)
1	1	20170913	23675	2413	25887	27	00010242fe8c5a6d1ba2dd792cb1...
2	2	20170426	95963	1661	10559	17	00018f77f2f0320c557190d7a144b...
3	3	20180114	38903	1978	11700	8	000229ec398224ef6ca0657da4fc7...
4	4	20180808	82958	834	17567	27	00024acbcdf0a6daa1e931b03811...
5	5	20170204	34382	2464	25588	17	00042b26cf59d7ce69dfabb4e55b4...

Gambar 4.14 Tabel fakta yang telah di-*load*

4.3 Analisis OLAP (*Online Analytical Processing*)

Setelah proses ETL selesai dan *data warehouse* berhasil dibangun, tahap selanjutnya adalah melakukan analisis multidimensi menggunakan pendekatan OLAP (*Online Analytical Processing*). Analisis ini bertujuan untuk menggali wawasan (*insight*) dari data transaksi pada *platform* Olist melalui berbagai perspektif, seperti waktu, kategori produk, lokasi, penjual (*seller*), dan metode pembayaran. Dengan menggunakan skema bintang (*star schema*) yang telah dirancang, proses eksplorasi data menjadi lebih mudah karena fakta penjualan telah terhubung dengan beberapa tabel dimensi relevan.

4.3.1. Analisis Penjualan Berdasarkan Waktu

Analisis berdasarkan waktu dilakukan dengan memanfaatkan dimensi tanggal (*dim_date*) untuk mengelompokkan transaksi berdasarkan bulan, kuartal, dan tahun. Hasil pengolahan OLAP menunjukkan adanya pola musiman (*seasonal pattern*) yang cukup jelas pada penjualan Olist.

```sql
SELECT
    d.year,
    SUM(f.price + f.freight_value) AS total_revenue,
    COUNT(f.order_id) AS total_orders
FROM dw.fact_sales f
JOIN dw.dim_date d ON f.date_key = d.date_key
GROUP BY d.year
ORDER BY d.year;
```

Gambar 4.15 *Query* penjualan per tahun

Secara umum, volume transaksi cenderung meningkat pada kuartal akhir (Oktober–Desember). Hal ini dapat dikaitkan dengan periode liburan, diskon akhir tahun, dan meningkatnya aktivitas belanja online. Sementara itu, beberapa bulan seperti Februari dan Maret cenderung mengalami penurunan, yang kemungkinan dipengaruhi oleh faktor ekonomi domestik maupun preferensi perilaku belanja masyarakat.

	year integer	total_revenue numeric	total_orders bigint
1	2016	59105.22	409
2	2017	7571228.92	54549
3	2018	9013397.16	64185

Gambar 4.16 Hasil penjualan per tahun

4.3.2. Analisis Kategori Produk Terlaris

Dimensi produk (*dim_product*) digunakan untuk mengevaluasi kategori produk yang paling banyak menghasilkan transaksi. Berdasarkan hasil OLAP, terdapat beberapa kategori yang konsisten menempati posisi teratas dari sisi volume penjualan maupun kontribusi *revenue*.

```sql
SELECT
    p.product_category_name,
    COUNT(f.order_item_id) AS total_items_sold
FROM dw.fact_sales f
JOIN dw.dim_product p ON f.product_key = p.product_key
GROUP BY p.product_category_name
ORDER BY total_items_sold DESC;
```

Gambar 4.17 *Query* penjualan per kategori produk

	product_category_name character varying (200)	total_items_sold bigint
1	cama_mesa_banho	11988
2	beleza_saude	10032
3	esporte_lazer	9004
4	moveis_decoracao	8832
5	informatica_acessorios	8150
6	utilidades_domesticas	7380
7	relogios_presentes	6213
8	telefonia	4726
9	ferramentas_jardim	4590
10	automotivo	4400

Gambar 4.18 Hasil penjualan per kategori produk

4.3.3. Analisis Performa *Seller*

Melalui pemanfaatan dimensi *seller* (*dim_seller*) dan tabel fakta penjualan, analisis OLAP menunjukkan adanya perbedaan performa yang cukup signifikan antar penjual pada *platform* Olist.

```
SELECT
    s.seller_id,
    s.seller_city,
    s.seller_state,
    SUM(f.price + f.freight_value) AS total_revenue
FROM dw.fact_sales f
JOIN dw.dim_seller s ON f.seller_key = s.seller_key
GROUP BY s.seller_id, s.seller_city, s.seller_state
ORDER BY total_revenue DESC;
```

Gambar 4.19 *Query* total penjualan per *seller*

	seller_id character varying (50)	seller_city character varying (100)	seller_state character varying (5)	total_revenue numeric
1	53243585a1d6dc2643021fd1853d8...	Lauro De Freitas	BA	258882.28
2	4b69f7a5dfa277a7dca6462dcf3b52...	Guariba	SP	258625.52
3	7c67e1448b00f6e969d365cea6b010...	Itaquaquecetuba	SP	254387.70
4	4a3ca9315b744ce9f8e93743614938...	Ibitinga	SP	252535.64
5	fa1c13f2614d7b5c4749cbc52fecda94	Sumare	SP	214454.82
6	da8622b14eb17ae2831f4ac5b9dab...	Piracicaba	SP	198621.24
7	7e93a43ef30c4f03f38b393420bc75...	Barueri	SP	189475.90
8	1025f0e2d44d7041d6cf58b6550e0b...	Sao Paulo	SP	178696.05
9	7a67c85e85bb2ce8582c35f2203ad7...	Sao Paulo	SP	172887.23
10	955fee9216a65b617aa5c0531780ce...	Sao Paulo	SP	163275.71

Gambar 4.20 Hasil total penjualan per *seller*

Hasil analisis mengungkap bahwa sebagian kecil seller memberikan kontribusi besar terhadap total pendapatan *platform* (*fenomena Pareto 80/20*). Selain itu, seller yang memiliki rating tinggi cenderung memiliki tingkat *repeat order* yang lebih baik, menunjukkan bahwa kualitas pelayanan sangat mempengaruhi keberlanjutan bisnis.

4.3.4. Analisis Metode Pembayaran

Dimensi pembayaran (*dim_payment*) digunakan untuk melihat perilaku pelanggan dalam memilih metode pembayaran. Dimensi pembayaran (*dim_payment*) digunakan untuk melihat perilaku pelanggan dalam memilih metode pembayaran. Hasil OLAP mengungkapkan bahwa pengguna lebih banyak menggunakan metode pembayaran *credit card*.

```
SELECT
    p.payment_type,
    COUNT(f.order_id) AS total_transactions
FROM dw.fact_sales f
JOIN dw.dim_payment p ON f.payment_key = p.payment_key
GROUP BY p.payment_type
ORDER BY total_transactions DESC;
```

Gambar 4.21 *Query* transaksi per metode pembayaran

	payment_type character varying (50)	total_transactions bigint
1	credit_card	87776
2	boleto	23190
3	voucher	6465
4	debit_card	1706
5	not_defined	3

Gambar 4.22 Hasil transaksi per metode pembayaran

4.3.5. Analisis Persebaran Pelanggan Berdasarkan Lokasi

Melalui dimensi geografi pelanggan (*dim_customer* dan kolom lokasi seperti *city* dan *state*), analisis OLAP dapat memetakan persebaran pelanggan pada skala kota dan negara bagian. Hasil analisis menunjukkan bahwa sebagian besar pelanggan berasal dari daerah dengan populasi padat seperti São Paulo (SP), Rio de Janeiro (RJ), Minas Gerais (MG), dan negara bagian di wilayah tenggara Brasil. Daerah-daerah ini memiliki infrastruktur logistik yang lebih baik serta aktivitas *e-commerce* yang tinggi.

```
SELECT
    c.customer_city,
    COUNT(DISTINCT c.customer_id) AS total_customers
FROM dw.fact_sales f
JOIN dw.dim_customer c ON f.customer_key = c.customer_key
GROUP BY c.customer_city
ORDER BY total_customers DESC;
```

Gambar 4.23 *Query* jumlah pelanggan per kota

	customer_city character varying (100)	total_customers bigint
1	Sao Paulo	15540
2	Rio De Janeiro	6882
3	Belo Horizonte	2773
4	Brasilia	2131
5	Curitiba	1521
6	Campinas	1444
7	Porto Alegre	1379
8	Salvador	1245
9	Guarulhos	1189
10	Sao Bernardo Do Campo	938

Gambar 4.24 Hasil jumlah pelanggan per kota

4.4. Interpretasi Hasil OLAP dan Evaluasi Teknis

Hasil OLAP tidak hanya menggambarkan pola penjualan, tetapi juga memberikan makna strategis bagi bisnis *e-commerce*. Peningkatan transaksi pada akhir tahun, misalnya, menunjukkan adanya pola musiman yang dapat dimanfaatkan untuk perencanaan promosi dan penguatan kapasitas operasional menjelang periode sibuk. Kategori produk yang konsisten mencatat penjualan tertinggi dapat menjadi dasar dalam manajemen stok, prioritas pemasaran, dan kerja sama dengan pemasok. Sementara itu, temuan bahwa sebagian kecil seller berkontribusi besar terhadap total penjualan menunjukkan perlunya strategi pembinaan seller berperforma rendah serta program insentif untuk seller unggulan agar kontribusinya tetap stabil. Interpretasi ini menegaskan bahwa *data warehouse* mampu membantu pengambilan keputusan yang lebih terarah.

Dari sisi teknis, penggunaan star schema terbukti meningkatkan efisiensi eksekusi query karena struktur yang sederhana dan jumlah join yang minimal. PostgreSQL mendukung pemrosesan paralel dan optimasi agregasi sehingga analisis dapat dijalankan dengan cepat meskipun data berasal dari beberapa tabel operasional. Namun, model ini memiliki keterbatasan, seperti cakupan analisis yang hanya berfokus pada proses penjualan serta kurangnya kedalaman analisis spasial akibat data geolokasi yang tidak sepenuhnya lengkap. Meskipun begitu, rancangan *data warehouse* yang dibangun tetap memberikan dasar yang kuat untuk analisis multidimensi dan dapat dikembangkan lebih lanjut sesuai kebutuhan.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil membangun *data warehouse e-commerce* berbasis PostgreSQL dengan *star schema* yang mampu mengintegrasikan data Olist dan mendukung analisis OLAP secara efisien. Hasil analisis menunjukkan pola penting seperti tren musiman penjualan, kategori produk paling diminati, serta kontribusi seller yang tidak merata. Secara teoretis, penelitian ini memberikan contoh penerapan pemodelan dimensional pada dataset *e-commerce* publik, sementara secara praktis, hasilnya dapat menjadi acuan bagi industri dalam pengembangan sistem analitik untuk perencanaan stok, evaluasi *seller*, dan strategi pemasaran.

Penelitian ini memiliki keterbatasan, terutama karena fokus pada satu fakta penjualan dan data geolokasi yang tidak sepenuhnya lengkap. Penelitian berikutnya dapat menambahkan fakta transaksi lain seperti logistik atau retur, serta mengembangkan integrasi *dashboard real-time* atau mekanisme pemuatan data *incremental* agar sistem *data warehouse* lebih relevan digunakan pada lingkungan industri maupun akademik.

DAFTAR PUSTAKA

[1] M. S. Farhan, A. Youssef, and L. Abdelhamid, "A Model for Enhancing Unstructured Big Data Warehouse Execution Time," *Big Data Cogn. Comput.*, vol. 8, no. 2, 2024, doi: 10.3390/bdcc8020017.

[2] A. Shahid, T. A. N. Nguyen, and M. T. Kechadi, "Big data warehouse for healthcare-sensitive data applications," *Sensors*, vol. 21, no. 7, 2021, doi: 10.3390/s21072353.

[3] L. Dinesh and K. G. Devi, "An efficient hybrid optimization of ETL process in data warehouse of cloud architecture," *J. Cloud Comput.*, vol. 13, no. 1, 2024, doi: 10.1186/s13677-023-00571-y.

[4] S. Fugkeaw, P. Suksai, and L. Hak, "SSF-CDW: achieving scalable, secure, and fast OLAP query for encrypted cloud data warehouse," *J. Cloud Comput.*, vol. 13, no. 1, 2024, doi: 10.1186/s13677-024-00692-y.

[5] A. Cakir, Ö. Akın, H. F. Deniz, and A. Yılmaz, "Enabling real time big data solutions for manufacturing at scale," *J. Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00672-6.

[6] S. Azzabi, Z. Alfughi, and A. Ouda, "Data Lakes: A Survey of Concepts and Architectures," *Computers*, vol. 13, no. 7, 2024, doi: 10.3390/computers13070183.

[7] T. Luhur, Rafika Rahmawati, Tri Puspa Rinjeni, Virdha Rahma Aulia, Prasasti Karunia Farista Ananto, and Iqbal Ramadhani Mukhlis, "Analyzing User Reviews with ETL using Pentaho Data Integration," *J. Komput. Teknol. Inf. Sist. Inf.*, vol. 4, no. 2, pp. 451–458, 2025, doi: 10.62712/juktisi.v4i2.403.

[8] J. M. Hesekiel, I. Arwani, and D. E. Ratnawati, "Pengembangan Data Warehouse untuk Evaluasi Pembelajaran Matakuliah Berdasarkan Data Kuesioner Mahasiswa di SIAM dan Rekapitulasi Presensi Dosen (Studi Kasus Teknologi Informasi Fakultas Ilmu Komputer)," *... Inf. dan Ilmu Komput. e-ISSN*, vol. 4, no. 7, pp. 2172–2177, 2020, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/7564/3589

[9] I. Thoib, B. P. Candra, F. B. Firmansyah, D. S. Nugraha, and N. Sururi, "Perancangan Data Warehouse Sebagai Penunjang Strategi Penerimaan Mahasiswa Baru," *J. Komput. Teknol. Inf. Sist. Inf.*, vol. 4, no. 1, pp. 98–108, 2025, doi: 10.62712/juktisi.v4i1.346.

[10] I. G. N. A. T. Putra, I. N. A. Mahendra, and I. M. S. Putra, "Implementasi ETL Data Warehouse Dengan Konsep Fitur Metadata Dan Cleansing Data," *J. Sist. Inf.*, vol. 9, no. 2, pp. 274–289, 2020.

[11] Y. Aisyah, S. Anwar, and - Samidi, "Pembuatan Data Warehouse secara Berjenjang dari Data Transaksi dengan ETL Script PHP," *Techno.Com*, vol. 22, no. 3, pp. 609–621, 2023, doi: 10.33633/tc.v22i3.8084.

[12] G. Bagus, A. Tama, N. Putu, M. Krisnayanti, P. Studi, and T. Informatika, "Penerapan Teknologi Datawarehouse Nosql Dan Business," vol. 3, no. 2, pp. 120–127, 2020.

[13] A. R. Nurridwan Firdaus and D. Firmansyah, "Implementasi Business Intelligence pada Data Pendapatan studi kasus (PT. Pos Indonesia)," *J. Esensi Infokom J. Esensi Sist. Inf. dan Sist. Komput.*, vol. 7, no. 2, pp. 33–39, 2023, doi: 10.55886/infokom.v7i2.686.