Jurnal Ilmiah Ilmu Komputer & Elektronika (MURAKOM)
Volume. 2 Nomor. 3 Juli 2026
e-ISSN : 3123-1985; p-ISSN : 3110-4754, Hal. 111-130
DOI: https://doi.org/10.64365/murakom.v2i3.340
Available online at: https://jurnal.muaraedukasi.id/index.php/murakom

Otomatisasi Pipeline ETL Menggunakan Machine Learning untuk Integrasi Data Heterogen pada Sektor Retail

Ismet Maulana Azhari[1*], Hafidz Ramadhan Ghiffari[2], Wensen Kollin[3] , Mujahid Izzudin Alqossam[4], Mohamad Hilman[5]
[1*-5]Universitas Sultan Ageng Tirtayasa, Indonesia
Alamat: Jl. Jendral Soedirman KM. 3, Cilegon, Banten 42435, Indonesia.
Email : 3337230014@untirta.ac.id[1], 3337230071@untirta.ac.id[2], 3337230081@untirta.ac.id[3], 3337230042@untirta.ac.id[4], mohamad.hilman@untirta.ac.id[5]
Korespondensi penulis: 3337230014@untirta.ac.id

Abstract. *This study discusses the automation of an Extract, Transform, Load (ETL) pipeline using Machine Learning for heterogeneous data integration in the retail sector. The main problem addressed in this study is the variation in format, structure, and attribute naming in retail operational data, especially in micro to medium-scale retail businesses. This condition causes data recapitulation and integration processes to rely heavily on manual work. This study designs an automated ETL pipeline with a focus on the data transformation stage using a Hybrid Schema Matching approach. This approach combines alias dictionary, token hints, value regex, and Machine Learning similarity based on TF-IDF Vectorization and Logistic Regression to standardize data attributes into a target schema. The expected result of this study is more consistent, structured, and ready-to-load data for a data warehouse to support Business Intelligence analysis. Therefore, the developed system is expected to improve the efficiency of retail data integration and reduce dependence on manual data transformation processes.*
Keywords: *Data Warehouse, ETL, Heterogeneous Data, Machine Learning, Retail, Schema Matching.*

Abstrak. Penelitian ini membahas otomatisasi pipeline Extract, Transform, Load (ETL) menggunakan Machine Learning untuk integrasi data heterogen pada sektor retail. Permasalahan utama yang diangkat adalah perbedaan format, struktur, dan penamaan atribut pada data operasional retail, khususnya pada retail berskala mikro hingga menengah. Kondisi tersebut menyebabkan proses rekapitulasi dan integrasi data masih banyak dilakukan secara manual. Penelitian ini merancang pipeline ETL otomatis dengan fokus pada tahap transformasi data menggunakan pendekatan Hybrid Schema Matching. Pendekatan ini menggabungkan alias dictionary, token hints, value regex, serta Machine Learning similarity berbasis TF-IDF Vectorization dan Logistic Regression untuk menstandarkan atribut data ke dalam skema target. Hasil penelitian ini diharapkan mampu menghasilkan data yang lebih konsisten, terstruktur, dan siap dimuat ke dalam data warehouse untuk mendukung kebutuhan analisis Business Intelligence. Dengan demikian, sistem yang dikembangkan dapat membantu meningkatkan efisiensi integrasi data retail dan mengurangi ketergantungan terhadap proses transformasi manual.
Kata kunci: Data Heterogen, Data Warehouse, ETL, Machine Learning, Retail, Schema Matching.

1. LATAR BELAKANG

Di era digital, data menjadi salah satu aset penting dalam mendukung pengambilan keputusan bisnis, termasuk pada sektor retail. Sistem data warehouse dan Business Intelligence (BI) memungkinkan pelaku usaha untuk memantau penjualan, mengelola persediaan, serta menyusun strategi bisnis berdasarkan data yang terintegrasi (Garani et al., 2019; Martins et al., 2020). Namun, pada retail berskala mikro hingga menengah seperti warung dan toko kelontong, pencatatan transaksi masih sering dilakukan secara manual dengan format yang tidak seragam.

Received: April 28, 2026; Revised: Mei 12, 2026; Accepted: Mei 30, 2026; Online Available: Juni 15, 2026; Published: Juli 13,2026

Permasalahan utama yang muncul adalah tingginya heterogenitas data. Setiap penjaga warung umumnya memiliki gaya penulisan, singkatan, dan format laporan yang berbeda-beda sehingga menghasilkan data semi-terstruktur yang sulit diolah secara langsung. Kondisi ini menyebabkan proses integrasi data menjadi tidak efisien dan masih memerlukan rekapitulasi manual. Padahal, integrasi data heterogen merupakan salah satu tantangan penting dalam pengelolaan data modern karena perbedaan struktur dan format dapat menghambat proses analisis data (Putrama & Martinek, 2024).

Untuk mengatasi permasalahan tersebut, proses Extract, Transform, Load (ETL) digunakan untuk mengintegrasikan data dari berbagai sumber ke dalam data warehouse. Akan tetapi, proses transformasi pada ETL konvensional masih banyak bergantung pada aturan manual seperti rule-based system atau regular expression. Pendekatan ini kurang efektif ketika menghadapi variasi bahasa yang dinamis dan tidak konsisten. Penelitian sebelumnya menunjukkan bahwa Machine Learning dapat membantu otomatisasi proses ETL dan integrasi data secara lebih adaptif (Mondal et al., 2020; Mondal & Saha, 2023). Selain itu, teknik Machine Learning juga dapat diterapkan untuk schema matching dalam menyelaraskan struktur data heterogen secara otomatis (Rodrigues & da Silva, 2021).

Meskipun berbagai penelitian telah membahas ETL dan integrasi data, masih terdapat keterbatasan pada penerapan pipeline ETL otomatis untuk data retail mikro yang bersifat tidak terstruktur dan menggunakan bahasa informal. Oleh karena itu, penelitian ini bertujuan untuk merancang pipeline ETL otomatis menggunakan Machine Learning untuk integrasi data heterogen pada sektor retail. Fokus penelitian berada pada proses transformasi data menggunakan Hybrid Schema Matching untuk menstandarisasi nama kolom dan struktur data sebelum dimuat ke dalam data warehouse. Dengan adanya sistem ini, proses integrasi data diharapkan menjadi lebih efisien, konsisten, dan siap digunakan untuk kebutuhan analisis Business Intelligence.

2. KAJIAN TEORITIS

Penelitian mengenai otomatisasi integrasi data dan pipeline ETL telah banyak dilakukan dalam beberapa tahun terakhir. Mondal dan Saha (2023) menjelaskan bahwa integrasi data pada data warehouse enterprise masih membutuhkan koordinasi manual yang tinggi, sehingga automated ETL diperlukan untuk menjaga kualitas dan ketersediaan data. Putrama dan Martinek (2024) juga menegaskan bahwa heterogenitas data, baik dari sisi semantik, struktur, maupun format, masih menjadi tantangan utama dalam integrasi data modern. Akan tetapi, kedua penelitian tersebut masih bersifat konseptual atau kajian literatur

dan belum menunjukkan implementasi empiris pada sektor retail dengan data heterogen secara nyata.

Beberapa penelitian lain secara khusus membahas pencocokan skema dan entitas pada data heterogen. Rodrigues dan da Silva (2021) membahas schema matching menggunakan Machine Learning dan bootstrapping untuk membangun data latih secara otomatis, sedangkan Jain et al. (2022) meneliti entity resolution berbasis active learning dan transformer untuk mencocokkan entitas dari berbagai sumber data. Kedua penelitian tersebut relevan dengan integrasi data retail karena data produk, pelanggan, dan transaksi sering memiliki atribut berbeda meskipun maknanya sama. Namun, fokus keduanya masih terbatas pada pencocokan skema atau entitas, belum pada pembangunan pipeline ETL secara menyeluruh.

Dari sisi arsitektur data warehouse, Vilela et al. (2023) mengusulkan arsitektur ETL real-time untuk integrasi data heterogen, sementara Medel (2026) membahas implementasi data warehouse dan OLAP pada sektor retail dengan sumber data seperti POS, CRM, dan cloud database. Solodovnikova et al. (2021) juga menunjukkan pentingnya kemampuan sistem dalam menyesuaikan perubahan skema data dari waktu ke waktu. Meskipun relevan dengan kebutuhan integrasi data retail, penelitian-penelitian tersebut belum menjadikan Machine Learning sebagai inti otomatisasi proses ETL.

Penelitian yang lebih dekat dengan otomatisasi berbasis kecerdasan buatan antara lain Kumar dan Kumar (2025), yang mengembangkan framework DeepETL berbasis deep learning dengan akurasi transformasi mencapai 92,4%, serta Mladenovic et al. (2026), yang membahas otomatisasi data preparation untuk Machine Learning. Namun, pendekatan tersebut belum secara khusus berfokus pada integrasi data retail mikro yang heterogen dan tidak terstruktur. Berdasarkan celah tersebut, penelitian ini berfokus pada pengembangan pipeline ETL otomatis berbasis Hybrid Schema Matching untuk meningkatkan proses integrasi, pemetaan skema, dan standarisasi data retail.

2.1 Data Warehouse

Data warehouse merupakan sistem penyimpanan terpusat yang dirancang untuk menampung data dari berbagai sumber operasional guna mendukung proses Business Intelligence dan pengambilan keputusan bisnis (Garani et al., 2019). Berbeda dengan sistem OLTP yang berfokus pada transaksi operasional harian, data warehouse menggunakan pendekatan OLAP yang dioptimalkan untuk analisis data historis dalam jumlah besar (Martins et al., 2020). Pada implementasinya, data warehouse umumnya menggunakan skema dimensional seperti star schema yang memisahkan tabel fakta dan tabel dimensi agar proses analisis data menjadi lebih efisien. Perkembangan teknologi cloud juga mendorong

penggunaan layanan seperti Google BigQuery dalam pembangunan data warehouse karena memiliki skalabilitas yang tinggi (Medel, 2026).

2.2 Pipeline ETL (*Extract, Transform, Load*)

ETL merupakan proses integrasi data yang terdiri atas tiga tahapan utama, yaitu extract, transform, dan load. Tahap extract dilakukan dengan mengambil data dari berbagai sumber operasional. Tahap transform dilakukan dengan membersihkan, memvalidasi, dan menyesuaikan format data agar sesuai dengan skema target. Sementara itu, tahap load dilakukan dengan memasukkan data hasil transformasi ke dalam data warehouse (Andriansyah, 2022). Dalam sistem modern, ETL menjadi komponen penting karena berfungsi sebagai penghubung antara data operasional dan sistem analitik (Dhaouadi et al., 2022). Namun, pendekatan ETL tradisional yang berbasis aturan masih memiliki keterbatasan ketika menghadapi data heterogen yang dinamis dan tidak terstruktur (Vilela et al., 2023).

2.3 Data Heterogen pada Sektor Retail

Data heterogen merupakan kumpulan data yang tidak memiliki keseragaman dalam format, struktur, maupun makna semantik. Pada sektor retail berskala mikro hingga menengah, pencatatan operasional harian sering dilakukan secara tidak terstandardisasi dan bergantung pada individu yang bertugas. Kondisi tersebut menghasilkan data semi-terstruktur maupun tidak terstruktur yang sulit diintegrasikan secara langsung ke dalam sistem analitik. Perbedaan struktur dan semantik data menjadi salah satu tantangan utama dalam proses integrasi data modern (Putrama & Martinek, 2024). Selain itu, perubahan format data dari waktu ke waktu juga menyebabkan proses integrasi menjadi lebih kompleks (Solodovnikova et al., 2021).

2.4 Machine Learning dalam Otomatisasi ETL

Penerapan Machine Learning pada pipeline ETL mengubah pendekatan transformasi data dari berbasis aturan menjadi berbasis model pembelajaran. Algoritma Machine Learning mampu mempelajari pola data secara otomatis tanpa harus diprogram untuk setiap variasi data yang muncul. Dalam tahap transformasi, Machine Learning dapat digunakan untuk mendeteksi anomali, mengenali variasi penulisan, serta melakukan pemetaan data tidak terstruktur ke dalam skema basis data secara otomatis (Mondal et al., 2020). Selain itu, penggunaan deep learning dalam proses ETL juga terbukti mampu meningkatkan akurasi transformasi data tidak terstruktur (Kumar & Kumar, 2025). Otomatisasi data preparation berbasis Machine Learning juga membantu menghasilkan data yang lebih bersih dan siap digunakan untuk analisis lanjutan (Mladenovic et al., 2026).

2.5 Schema Matching dan Hybrid Schema Matching

Schema matching merupakan proses pencarian kesamaan semantik antar atribut atau skema data yang berbeda (Rodrigues & da Silva, 2021). Dalam integrasi data retail, schema matching diperlukan karena data dari berbagai sumber sering memiliki nama atribut berbeda meskipun memiliki makna yang sama. Penelitian ini menggunakan pendekatan Hybrid Schema Matching Engine untuk meningkatkan kemampuan pencocokan atribut data heterogen. Pendekatan tersebut menggabungkan beberapa metode seperti Alias Dictionary, Token Hints, Value Regex, serta Machine Learning Similarity menggunakan TF-IDF Vectorization dan Logistic Regression. Pendekatan hybrid digunakan agar sistem lebih adaptif dalam menangani variasi penulisan data retail yang tidak konsisten.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan berbasis rekayasa sistem (*system engineering approach*) dengan model pengembangan berbasis event-driven architecture (EDA). Pendekatan tersebut dipilih karena mampu mendukung pemrosesan data secara otomatis, real-time, dan scalable pada lingkungan data retail yang memiliki karakteristik heterogen baik dari sisi struktur maupun penamaan atribut data. Sistem yang dibangun dirancang untuk mampu melakukan proses extract, transform, dan load (ETL) secara otomatis terhadap data CSV dengan skema yang berbeda-beda menggunakan metode hybrid schema matching berbasis machine learning.

3.1 Arsitektur dan Alur Kerja Pipeline ETL

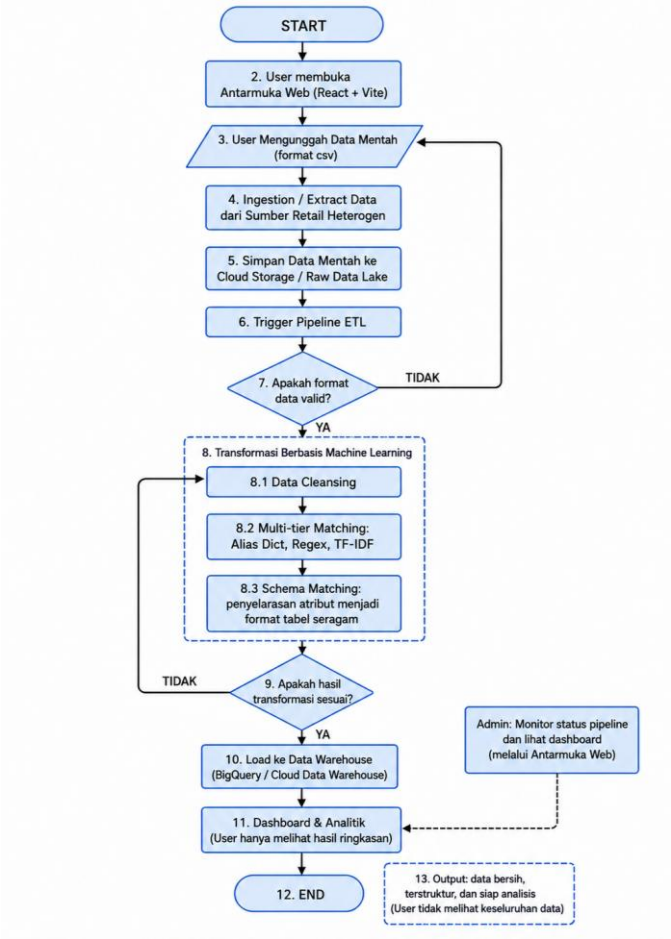
Metodologi yang digunakan dalam pembangunan Data Warehouse ini berbasis pada pendekatan *Event-Driven Architecture* (EDA). Pipeline dirancang untuk memproses berkas *Comma-Separated Values* (CSV) yang heterogen secara otomatis dan *real-time* begitu berkas diunggah oleh pengguna. Alur kerja sistem secara menyeluruh terbagi menjadi beberapa tahapan utama, mulai dari interaksi pengguna di sisi antarmuka hingga penyimpanan akhir di data warehouse cloud.

Tabel 1. Tahapan dan Deskripsi Alur Operasional Sistem

No	Tahapan/ Komponen	Deskripsi Alur Operasional Sistem

1	User Upload & Frontend	Pengguna berinteraksi dengan aplikasi *Single Page Application* (SPA) yang dibangun menggunakan React 18.2 dan Vite 5.0. Melalui antarmuka dasbor kustom ini, pengguna mengunggah berkas CSV mentah yang memiliki skema kolom tidak seragam.
2	Backend Gateway	Berkas CSV yang diunggah diterima oleh API Gateway berbasis FastAPI (Python). Backend ini bertanggung jawab menangani sesi autentikasi pengguna serta memvalidasi ukuran dan format awal berkas sebelum diteruskan ke penyimpanan cloud.
3	Raw Landing Bucket	FastAPI mengunggah berkas CSV mentah tersebut ke dalam wadah penyimpanan objek cloud, yaitu Google Cloud Storage (GCS), spesifik pada bagian *Raw Landing Bucket*. Berkas disimpan dalam format aslinya sebagai data historis mentah.
4	Event-Driven Trigger	Setiap ada berkas baru yang masuk ke *Raw Landing Bucket*, layanan Google Cloud Eventarc akan menangkap *event* tersebut secara otomatis dan instan (*real-time*). Eventarc kemudian mengirimkan sinyal pemicu (*trigger HTTP POST request*) ke modul kecerdasan buatan.
5	Cloud Run ML Engine	Sinyal pemicu dari Eventarc diterima oleh modul Machine Learning yang dideploy di atas Google Cloud Run dengan teknologi kontainerisasi Docker. Di dalam lingkungan serverless ini, mesin pemetaan skema (*Schema Matching Engine*) dieksekusi secara terisolasi dan terukur.
6	Hybrid Schema Matching	Modul ML melakukan transformasi data melalui strategi pencocokan skema 4-tier (*Multi-Tier Matching*) untuk menyelaraskan nama kolom input menjadi 5 atribut standar target, yaitu: product_id, product_name, price, stock, dan category. Langkah pemetaan ini meliputi: 1. Alias Dictionary: Kamus sinonim kata/istilah lokal dan singkatan.

		2. Token Hints: Analisis kedekatan kata kunci semantik. 3. Value Regex: Pemeriksaan pola nilai baris data untuk validasi tipe data. 4. ML Similarity: Pembobotan tekstual menggunakan *TF-IDF Vectorization* dan klasifikasi memakai algoritma *Logistic Regression*.
7	Data Loading to DW	Setelah skema kolom diselaraskan dan data dinyatakan bersih (*clean data*), modul mengirimkan data tersebut ke Google BigQuery sebagai Data Warehouse utama. Data dimasukkan ke dalam tabel fakta retail yang terstruktur untuk kebutuhan analitik.
8	Local Fallback Mechanism	Sebagai bentuk penanganan kegagalan sistem (*fault tolerance*) untuk menjamin penyediaan layanan yang andal (*High Availability*), sistem dilengkapi dengan mekanisme cadangan lokal. Jika koneksi atau API menuju Google BigQuery mengalami kendala/*error*, data hasil transformasi akan dialihkan dan disimpan sementara ke dalam direktori lokal backend dengan penamaan berkas processed_data_local.csv.
9	Interactive Analytics Dashboard	Data terstruktur yang telah tersimpan di Google BigQuery diakses kembali oleh antarmuka React Dashboard melalui REST API FastAPI. Komponen dasbor kustom (seperti Dashboard.jsx) melakukan visualisasi metrik bisnis secara interaktif, real-time, dan multi-role langsung pada aplikasi web tanpa ketergantungan pada alat pihak ketiga seperti Looker Studio.



Gambar 1. Flowchart *Pipeline ETL* Otomatis

Berdasarkan Gambar di atas, alur kerja sistem dimulai secara linear dari sisi kiri (interaksi pengguna) menuju ke sisi kanan (pemrosesan data oleh mesin ML dan penyimpanan). Secara spesifik, bagan alir ini menegaskan karakteristik High Availability pada sistem, di mana setelah fase Hybrid Schema Matching selesai dieksekusi di Google Cloud Run, terdapat percabangan kondisi (*decision gate*). Jika gerbang utama menuju Google BigQuery aktif, data langsung dimuat ke warehouse. Namun, jika terjadi kegagalan jaringan cloud, aliran data secara otomatis dialihkan ke jalur Local Fallback (*processed_data_local.csv*) sebelum akhirnya divisualisasikan ke dalam React Dashboard.

3.2 Instrumen dan Lingkungan Pengembangan

Untuk memastikan performa pipeline data berjalan optimal, terstandarisasi, dan mudah di deploy, lingkungan pengembangan dikonfigurasi menggunakan instrumen perangkat lunak modern sebagai berikut:

a. Bahasa Pemrograman & Framework Utama: Python 3.10+ (sisi backend dan ML) dengan framework FastAPI untuk performa tinggi asinkronus, serta

JavaScript/TypeScript dengan React 18.2 untuk sisi antarmuka pengguna.

b. Teknologi Orkestrasi & Cloud Native: Google Cloud Eventarc sebagai agen penggerak event otomatis, Google Cloud Storage sebagai media penyimpanan objek, dan Google Cloud Run untuk komputasi nirserver.
c. Kontainerisasi: Docker digunakan untuk membungkus kode backend dan mesin Machine Learning ke dalam bentuk *image* mandiri. Hal ini menjamin konsistensi eksekusi lingkungan sistem baik pada saat tahap pengembangan lokal maupun saat naik ke tahap produksi cloud.
d. Dataset: Dataset pengujian secara eksklusif menggunakan dataset publik dari platform Kaggle, yaitu *Indonesian Micro-Retail Heterogeneous Sales Data*. Dataset ini digunakan sebagai sumber data mentah utama guna menyimulasikan tingginya variasi format dan heterogenitas penamaan kolom pada pencatatan operasional retail skala mikro di Indonesia secara nyata.
e. Infrastruktur CI/CD: Proses integrasi dan pengiriman kode otomatis dikelola menggunakan layanan GitHub Actions. Setiap perubahan kode pada repositori akan memicu pengujian otomatis dan otomatisasi pembaruan container di Google Cloud Platform tanpa waktu henti (*zero downtime deployment*).
f. Mesin Penyimpanan Data (Data Storage): Google BigQuery difungsikan sebagai repositori pusat data retail berskala besar (olap) yang mendukung query SQL analitis secara cepat.
g. Keamanan dan Manajemen Kredensial: Untuk memastikan keamanan akses tingkat arsitektur, sistem mengadopsi prinsip pemisahan konfigurasi (berdasarkan Twelve-Factor App). Seluruh kredensial sensitif, seperti kunci autentikasi multi-peran (berbasis JSON Web Token) serta parameter kunci otorisasi layanan Google Cloud, diisolasi ke dalam variabel lingkungan (Environment Variables). Hal ini mencegah terekspos nya hardcoded credentials pada repositori kode (melalui audit gitignore), sekaligus memberikan sistem perlindungan keamanan berlapis pada peladen produksi.

3.3 Teknik Analisis dan Evaluasi Sistem

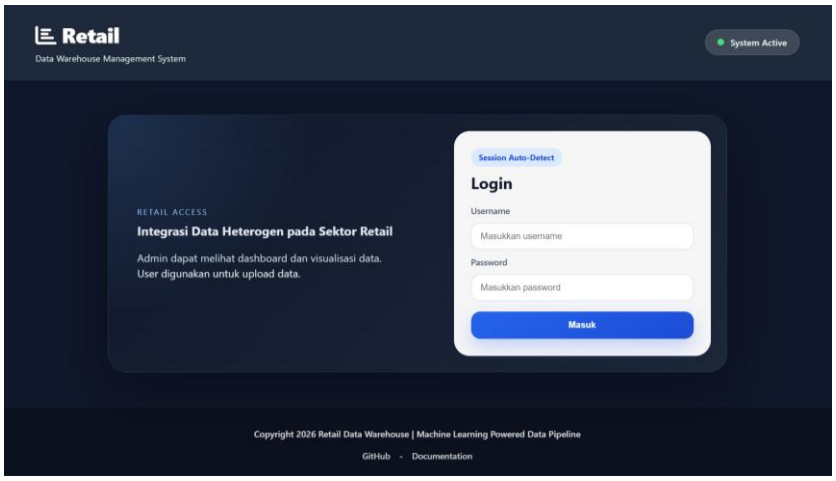
Untuk memastikan kelayakan dan menyatakan bahwa sistem ini berhasil, dilakukan validasi melalui dua pengujian utama:

a. Uji Fungsionalitas: Evaluasi untuk memastikan seluruh alur pipeline ETL (mulai dari ekstraksi hingga pemuatan ke Data Warehouse) dapat berjalan secara otomatis tanpa terjadi hambatan atau error.
b. Uji Performa: Mengukur tingkat efisiensi dan akurasi sistem. Evaluasi performa pipeline

diukur menggunakan Execution Time Analysis untuk melihat waktu yang dibutuhkan dalam memproses data. Sementara itu, performa model *Machine Learning* diuji berdasarkan nilai akurasi klasifikasi dan *matching rate* mesin pemeta skema dalam mengenali variasi nama kolom baru.

4. HASIL DAN PEMBAHASAN

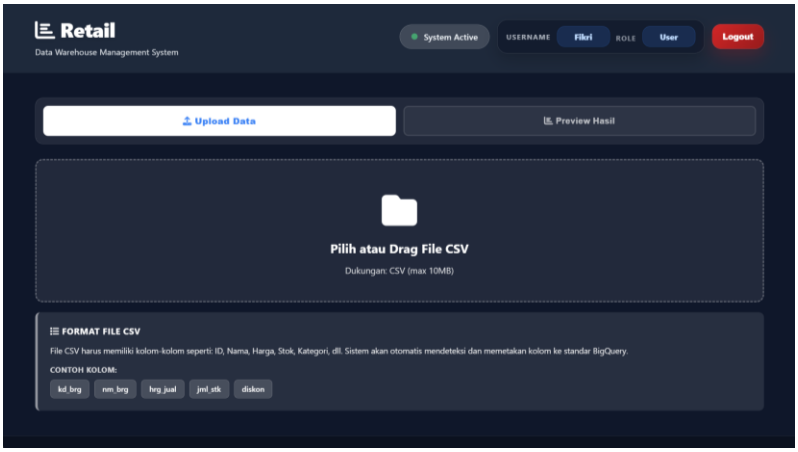
4.1 Hasil Implementasi Sistem



Gambar 2. Login Page

Gambar 2 menunjukkan tampilan halaman login pada aplikasi Retail Data Warehouse yang telah dikembangkan. Halaman ini berfungsi sebagai gerbang autentikasi pengguna sebelum mengakses fitur utama sistem. Pengguna diwajibkan memasukkan username dan password yang valid agar dapat masuk ke dalam sistem sesuai dengan hak akses masing-masing. Implementasi autentikasi ini bertujuan untuk menjaga keamanan data retail dan membatasi akses berdasarkan role pengguna.

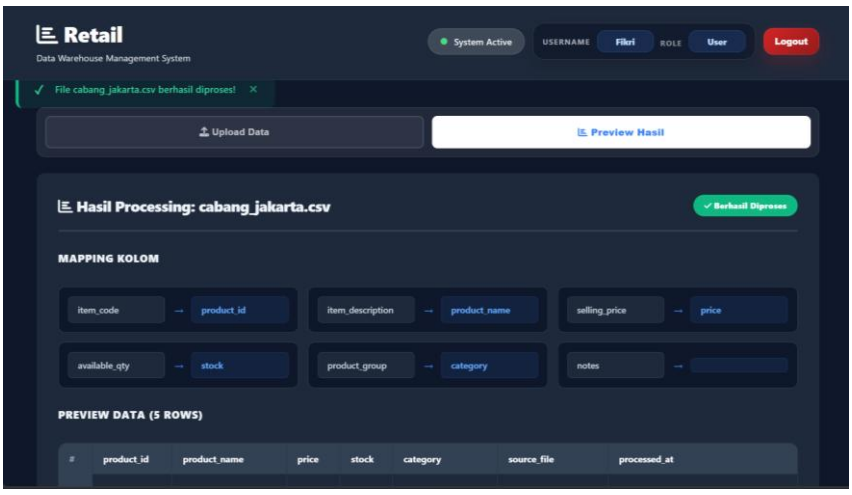
Pada implementasinya, sistem menerapkan mekanisme multi-role access yang terdiri atas Admin dan User. Role Admin memiliki akses untuk melihat dashboard analitik dan visualisasi data warehouse, sedangkan role User difokuskan untuk melakukan proses upload data CSV ke dalam pipeline ETL. Selain itu, pada halaman login juga ditampilkan indikator status sistem (“System Active”) yang menunjukkan bahwa layanan backend dan pipeline data sedang berjalan secara aktif. Antarmuka dibangun menggunakan React 18.2 dan Vite 5.0 sehingga halaman dapat berjalan secara responsif dan ringan pada browser pengguna.



Gambar 3. Page Upload File Data

Gambar 3 menunjukkan halaman upload data yang digunakan oleh User untuk memasukkan file data retail ke dalam sistem. Pada halaman ini, pengguna dapat memilih atau melakukan drag file CSV ke area unggah yang telah disediakan. File yang diunggah kemudian menjadi input awal dalam proses ETL sebelum masuk ke tahap ekstraksi, transformasi, dan pemuatan data ke data warehouse.

Pada halaman ini juga ditampilkan informasi mengenai format file yang didukung, batas ukuran file, serta contoh nama kolom seperti kd_brg, nm_brg, hrg_jual, jml_stk, dan diskon. Informasi tersebut membantu pengguna memahami bentuk data yang dapat diproses oleh sistem. Setelah file berhasil diunggah, sistem akan mendeteksi struktur kolom secara otomatis dan memetakan kolom tersebut ke format standar BigQuery agar data dapat dianalisis secara lebih konsisten.



Gambar 4. Tampilan Setelah File Diupload

Gambar 4 menunjukkan tampilan sistem setelah file cabang_jakarta.csv berhasil diunggah dan diproses. Pada halaman ini, sistem menampilkan notifikasi bahwa file berhasil diproses, sehingga pengguna dapat mengetahui bahwa proses upload, ekstraksi, dan transformasi data telah berjalan tanpa error.

Bagian utama pada halaman ini menampilkan hasil mapping kolom, yaitu proses pemetaan nama kolom dari file sumber ke format standar data warehouse. Contohnya, kolom item_code dipetakan menjadi product_id, item_description menjadi product_name, selling_price menjadi price, available_qty menjadi stock, dan product_group menjadi category. Hasil ini menunjukkan bahwa sistem mampu mengenali variasi nama kolom dari file CSV heterogen dan menyesuaikannya dengan struktur standar yang digunakan pada Google BigQuery.

PREVIEW DATA (5 ROWS)

#	product_id	product_name	price	stock	category	source_file	processed_at
1	JKT-001	Beras Premium 5Kg	64500	45	Food & Beverage	cabang_jakarta.csv	2026-05-27T15:27:34.668053+00:00
2	JKT-002	Minyak Goreng 2L	31500	12	Daily Needs	cabang_jakarta.csv	2026-05-27T15:27:34.668053+00:00
3	JKT-003	Sabun Mandi Cair	24000	80	Personal Care	cabang_jakarta.csv	2026-05-27T15:27:34.668053+00:00
4	JKT-004	Susu Uht 1L	18000	20	Food & Beverage	cabang_jakarta.csv	2026-05-27T15:27:34.668053+00:00
5	JKT-005	Roti Tawar	15000	15	Food & Beverage	cabang_jakarta.csv	2026-05-27T15:27:34.668053+00:00

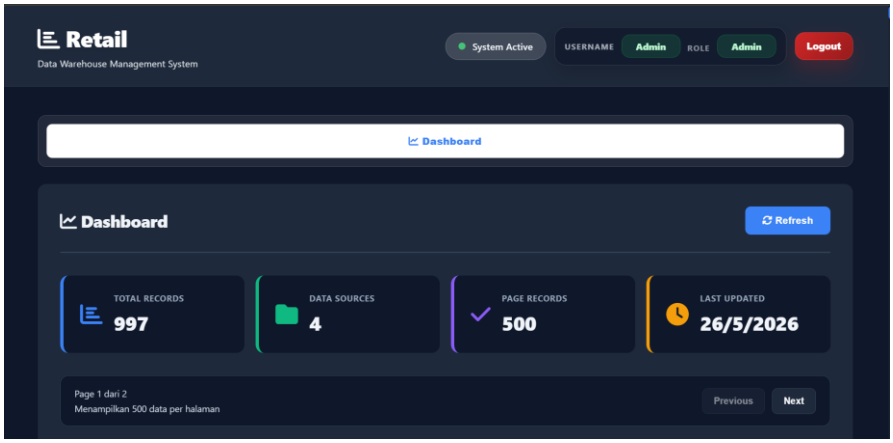
Gambar 5. Tabel Analisis

Gambar 5 menunjukkan tampilan preview data setelah file berhasil diunggah dan diproses oleh sistem. Pada tampilan ini, User hanya dapat melihat lima baris pertama dari data yang telah dimasukkan. Pembatasan ini diterapkan karena role User difokuskan pada proses input dan validasi awal data, bukan pada analisis keseluruhan data warehouse.

Preview data menampilkan kolom standar hasil transformasi, yaitu product_id, product_name, price, stock, category, source_file, dan processed_at. Kolom-kolom tersebut menunjukkan bahwa data dari file cabang_jakarta.csv telah berhasil diseragamkan ke dalam format yang siap digunakan untuk proses analisis. Berbeda dengan User, Admin memiliki akses untuk melihat data secara lebih lengkap melalui dashboard dan tampilan data warehouse. Dengan demikian, fitur ini menunjukkan bahwa sistem telah menerapkan pembatasan akses berdasarkan role pengguna.

Untuk memperjelas perbedaan hak akses tersebut, antarmuka khusus Admin dirancang untuk menyediakan visibilitas menyeluruh terhadap performa operasional toko dan repositori data warehouse. Saat pertama kali masuk, Admin akan disambut oleh halaman utama dasbor

yang merangkum metrik bisnis penting secara real-time. Tampilan dashbord utama pada sisi Admin ini ditunjukkan pada Gambar 6.



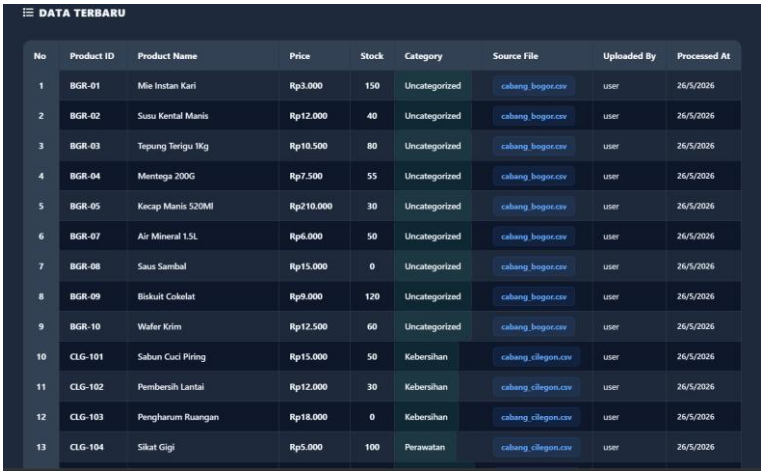
Gambar 6. Tampilan Antarmuka Dashbord Admin

Guna mendukung proses pengambilan keputusan strategis (*Data-Driven Decision Making*), Admin dapat mengakses menu visualisasi lanjutan yang terhubung langsung ke Google BigQuery Data Warehouse. Halaman ini menyajikan grafik analitik performa penjualan harian, tren inventaris, serta distribusi kategori produk secara interaktif guna mempermudah pemilik retail dalam menganalisis data secara mendalam. Tampilan grafik analitik data warehouse ini diperlihatkan pada Gambar 7.

Gambar 7. Tampilan Grafis Analitik dan Visualisasi pada Sisi Admin

Selain ringkasan metrik utama dan visualisasi, sistem juga menyediakan halaman pengelolaan berkas terpusat (*Data Management UI*). Melalui fitur ini, Admin dapat memantau seluruh riwayat unggahan berkas CSV dari berbagai cabang kasir retail, memeriksa log

aktivitas, serta melihat status keberhasilan transformasi skema yang dijalankan oleh engine Machine Learning. Tampilan halaman manajemen data tersebut disajikan pada Gambar 8.



Gambar 8. Tampilan Manajemen Berkas dan Riwayat Sinkronisasi ETL

Melalui integrasi kontrol akses dan ketiga subsistem antarmuka ini, sistem memastikan bahwa pengguna kasir biasa hanya dibatasi pada hak akses pengunggahan dan pencocokan sementara (*preview*), sedangkan hak otorisasi untuk pengawasan alur ETL, manajemen berkas historis, dan analisis visual menyeluruh tetap terisolasi dengan aman di bawah peran Admin.

4.2 Hasil Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur pada pipeline ETL berbasis machine learning dapat berjalan sesuai dengan fungsi yang telah dirancang. Pengujian dilakukan terhadap fitur utama sistem, mulai dari proses autentikasi pengguna, upload file CSV, schema matching, transformasi data, hingga proses loading data ke data warehouse dan visualisasi dashboard.

Tabel 2. Hasil Pengujian Sistem

No	Fitur/Komponen yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
1	Login Pengguna	Admin dan User masuk menggunakan akun yang valid	Sistem berhasil mengarahkan pengguna ke halaman sesuai role masing-masing	Pengguna berhasil masuk ke dashboard sesuai role yang dimiliki	Berhasil

2	Validasi File Upload	Pengguna mengunggah file berformat CSV	Sistem menerima file CSV dan melanjutkan proses ekstraksi data	File CSV berhasil diterima oleh sistem	Berhasil
3	Penolakan Format File	Pengguna mengunggah file selain CSV	Sistem menolak file dan menampilkan pesan kesalahan	Sistem menampilkan pesan bahwa format file tidak sesuai	Berhasil
4	Preview Data Mentah	File CSV berhasil diunggah ke sistem	Sistem menampilkan sebagian isi data mentah dalam bentuk tabel preview	Data mentah berhasil ditampilkan pada halaman preview	Berhasil
5	Ekstraksi Data	Sistem membaca isi file CSV yang telah diunggah	Data dari file CSV dapat dibaca tanpa menyebabkan error pada sistem	Data berhasil diekstraksi dan diteruskan ke tahap transformasi	Berhasil
6	Handling Encoding File	Sistem menerima file CSV dengan encoding yang berbeda	Sistem tetap mampu membaca file menggunakan fallback encoding	File CSV berhasil dibaca meskipun memiliki format encoding yang tidak standar	Berhasil

7	Schema Matching	Sistem menerima kolom tidak seragam seperti nama_barang, harga_jual, atau stok_produk	Sistem memetakan kolom tersebut ke atribut standar data warehouse	Kolom berhasil dipetakan ke atribut standar seperti product_name, price, dan stock	Berhasil
8	Data Cleansing	Data memiliki nilai kosong, format angka tidak seragam, atau karakter yang tidak sesuai	Sistem membersihkan dan menyeragamkan data sebelum disimpan	Data berhasil dinormalisasi dan siap dimuat ke data warehouse	Berhasil
9	Loading ke BigQuery	Data hasil transformasi dikirim ke Google BigQuery	Data bersih tersimpan pada tabel tujuan di data warehouse	Data berhasil dimuat ke Google BigQuery	Berhasil
10	Local Fallback	Koneksi ke Google BigQuery mengalami gangguan	Sistem menyimpan data hasil transformasi ke penyimpanan lokal	File cadangan processed_data_local.csv berhasil terbentuk	Berhasil
11	API Dashboard	Frontend meminta data hasil ETL melalui backend	Backend mengirimkan data hasil ETL ke frontend	Data berhasil diterima dan ditampilkan pada dashboard	Berhasil

12	Dashboard Analitik	Data warehouse telah terisi data hasil ETL	Dashboard menampilkan metrik dan visualisasi data retail	Grafik dan ringkasan data berhasil ditampilkan secara interaktif	Berhasil
13	Server-Side Pagination	Dashboard menampilkan data dalam jumlah besar	Sistem membatasi data yang ditampilkan menggunakan limit dan offset	Data berhasil ditampilkan secara bertahap tanpa membebani halaman	Berhasil
14	Multi-Role Access	Admin dan User mengakses fitur sistem	Hak akses ditampilkan sesuai peran masing-masing pengguna	Admin dan User memiliki tampilan serta akses fitur yang berbeda	Berhasil
15	Error Handling File Bermasalah	Pengguna mengunggah file kosong atau rusak	Sistem memberikan peringatan tanpa menyebabkan aplikasi berhenti	Sistem menampilkan pesan error dan tidak mengalami crash	Berhasil

Berdasarkan hasil pengujian pada Tabel 2, seluruh fitur utama sistem berhasil dijalankan sesuai dengan hasil yang diharapkan. Sistem mampu melakukan proses upload file CSV, membaca data heterogen, melakukan mapping kolom secara otomatis, serta memuat data ke dalam Google BigQuery tanpa mengalami kegagalan sistem. Selain itu, mekanisme fallback lokal juga berhasil berjalan ketika koneksi cloud mengalami gangguan. Hasil tersebut menunjukkan bahwa pipeline ETL yang dibangun telah mampu mendukung otomatisasi integrasi data retail secara berjalan dengan baik.

4.3 Analisis Manfaat dan Capaian Proyek

Berdasarkan hasil pengujian fungsionalitas pipeline yang telah diimplementasikan, sistem menunjukkan beberapa capaian utama dalam mendukung proses integrasi data retail.

Sistem mampu mengurangi pekerjaan manual karena pengguna tidak perlu lagi menyamakan format kolom dari berbagai file CSV secara satu per satu. Proses upload, pemetaan kolom, transformasi, dan penyimpanan data dapat berjalan secara otomatis melalui pipeline ETL berbasis Machine Learning.

Selain itu, sistem juga mampu menjaga konsistensi data dengan menyeragamkan format data ke dalam struktur standar Data Warehouse. Fitur fallback lokal turut meningkatkan keandalan sistem karena data tetap dapat disimpan sementara ketika koneksi cloud mengalami gangguan. Dengan adanya dashboard analitik, Admin dapat melihat hasil pengolahan data secara lebih cepat untuk mendukung pemantauan penjualan, stok, dan performa retail.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, sistem "Otomatisasi Pipeline ETL Menggunakan Machine Learning untuk Integrasi Data Heterogen pada Sektor Retail" berhasil dibangun sebagai solusi integrasi data retail berbasis pipeline ETL otomatis. Sistem ini mampu menghubungkan proses upload data, pemrosesan skema, transformasi, penyimpanan ke Google BigQuery, serta penyajian data melalui dashboard berbasis React. Penerapan Event-Driven Architecture dengan Google Cloud Eventarc juga memungkinkan proses pemrosesan data berjalan secara otomatis setelah file diunggah, tanpa perlu penjadwalan manual.

Komponen Hybrid Schema Matching yang menggabungkan Alias Dictionary, Token Hints, Value Regex, TF-IDF Vectorization, dan Logistic Regression berhasil membantu sistem dalam mengenali variasi nama kolom dari data retail yang heterogen. Kolom-kolom dari file sumber dapat dipetakan ke atribut standar data warehouse, yaitu product_id, product_name, price, stock, dan category. Selain itu, fitur local fallback juga meningkatkan keandalan sistem karena data hasil transformasi tetap dapat disimpan sementara ketika koneksi ke Google BigQuery mengalami gangguan.

Untuk pengembangan selanjutnya, sistem dapat ditingkatkan dengan menerapkan model pencocokan skema berbasis Deep Learning atau Transformer agar mampu mengenali variasi nama kolom yang lebih kompleks. Selain itu, format input juga dapat diperluas agar tidak hanya mendukung file CSV, tetapi juga JSON, XML, dan XLSX. Pengembangan fitur notifikasi otomatis melalui email atau Telegram juga dapat ditambahkan agar admin segera mengetahui apabila terjadi kegagalan pipeline atau saat mekanisme fallback aktif.

DAFTAR REFERENSI

Andriansyah, D. (2022). Implementasi extract-transform-load (ETL) data warehouse laporan harian pool. *Jurnal Teknologi Informasi*. https://ejournal.antarbangsa.ac.id/jti/article/view/486

Dhaouadi, A., Bousselmi, K., Gammoudi, M. M., Monnet, S., & Hammoudi, S. (2022). Data warehousing process modeling from classical approaches to new trends: Main features and comparisons. *Data, 7*(8), 113. https://doi.org/10.3390/data7080113

Garani, G., Chernov, A., Savvas, I., & Butakova, M. (2019). A data warehouse approach for business intelligence. *Proceedings of the IEEE*. https://ieeexplore.ieee.org/abstract/document/8795395

Gonzales, R., Wareham, J., & Serida, J. (2015). Measuring the impact of data warehouse and business intelligence on enterprise performance in Peru: A developing country. *Journal of Global Information Technology Management, 18*(3), 162–187. https://doi.org/10.1080/1097198X.2015.1070616

Jain, A., Sarawagi, S., & Sen, P. (2022). Deep indexed active learning for matching heterogeneous entity representations. *Proceedings of the VLDB Endowment, 15*(1), 31–45. https://doi.org/10.14778/3485450.3485455

Kumar, G. S. S., & Kumar, M. R. (2025). Deep learning-based ETL framework for automating data transformation in big data analytics. *Mathematical Modelling of Engineering Problems, 12*(11), 3971–3979. https://doi.org/10.18280/mmep.121123

Maniar, V., Reddy, R. K., Rajendran, D., Namburi, V. D., Singh, A. A., & Tamilmani, V. (2021). Review of streaming ETL pipelines for data warehousing: Tools, techniques, and best practices. *International Journal of AI, Big Data, Computational and Management Studies*. https://ijaibdcms.org/index.php/ijaibdcms/article/view/284/

Martins, A., Martins, P., Caldeira, F., & Sá, F. (2020). An evaluation of how big-data and data warehouses improve business intelligence decision making. In *Advances in Intelligent Systems and Computing*. Springer. https://doi.org/10.1007/978-3-030-45688-7_61

Medel, V. P. (2026). A reproducible data warehouse and OLAP framework for retail analytics: Design, dimensional modeling, and experimental evaluation in the SwiftMart case. *International Journal of Information Technology and Computer Science Applications, 4*(1), 35–46. https://ejurnal.jejaringppm.org/index.php/jitcsa/article/download/244/181

Mladenovic, S., Lindauer, M., & Doerr, C. (2026). Automated data preparation for machine learning: A survey. *Journal of Data-centric Machine Learning Research*, *3*(2), 1–72.

https://openreview.net/forum?id=Euti6LHIOs

Mondal, K. C., Biswas, N., & Saha, S. (2020). Role of machine learning in ETL automation. In *Proceedings of the 21st International Conference on Distributed Computing and Networking* (Article 57, pp. 1–6). https://doi.org/10.1145/3369740.3372778

Mondal, K. C., & Saha, S. (2023). Data integration process automation using machine learning: Issues and solution. In L. Rokach, O. Maimon, & E. Shmueli (Eds.), *Machine learning for data science handbook* (pp. 39–54). Springer. https://doi.org/10.1007/978-3-031-24628-9_3

Putrama, I. M., & Martinek, P. (2024). Heterogeneous data integration: Challenges and opportunities. *Data in Brief, 56*, Article 110853. https://doi.org/10.1016/j.dib.2024.110853

Rodrigues, D., & da Silva, A. (2021). A study on machine learning techniques for the schema matching network problem. *Journal of the Brazilian Computer Society, 27*, Article 14. https://doi.org/10.1186/s13173-021-00119-5

Solodovnikova, D., Niedrite, L., & Svilpe, L. (2021). Managing evolution of heterogeneous data sources of a data warehouse. In *Proceedings of the 23rd International Conference on Enterprise Information Systems* (Vol. 1, pp. 105–117). https://doi.org/10.5220/0010496601050117

Vilela, F. de A., Times, V. C., Bernardi, A. C. de C., Freitas, A. de P., & Ciferri, R. R. (2023). A non-intrusive and reactive architecture to support real-time ETL processes in data warehousing environments. *Heliyon, 9*(5), Article e15728. https://doi.org/10.1016/j.heliyon.2023.e15728