



Perancangan dan Implementasi Arsitektur Data Pipeline Otomatis untuk Analisis Sentimen Ulasan Aplikasi E-Commerce menggunakan Apache Airflow, Docker, dan Ensemble Learning

M. Shandy Agustian^{1*}, Fidi Supriadi², David Setiadi³

¹⁻³Informatika, Universitas Sebelas April, Indonesia

*Penulis Korespondensi: shandyagustian0@gmail.com

Abstract. *The growth of user reviews on e-commerce platforms in Indonesia is growing much faster than conventional analysis capacity that relies on manual review. This study designs and implements an automated data pipeline based on an ETL (Extract, Transform, Load) architecture that integrates data acquisition through dynamic web scraping (Selenium), Indonesian text preprocessing, sentiment classification using Ensemble Learning models (Multinomial Naive Bayes and Random Forest), workflow orchestration using Apache Airflow, containerization with Docker, and storage of results in a PostgreSQL database. The system is tested using a case study of reviews of three popular e-commerce applications in Indonesia (Shopee, Tokopedia, and Blibli) with a total of more than 30,000 rows of raw review data. The test results show that the pipeline runs successfully end-to-end and automatically, with an idempotency mechanism that successfully maintains data integrity from the risk of duplication. The Ensemble model achieves an overall accuracy of 77.94% using the SMOTE (Synthetic Minority Over-sampling Technique) technique to address class imbalance. However, a per-class evaluation analysis revealed that SMOTE was only effective in improving performance on minority classes with relatively sufficient source data, while failing to provide significant improvements on classes with extreme imbalance. This finding provides a methodological contribution regarding the limitations of oversampling techniques' effectiveness in very limited data and emphasizes the importance of evaluating per-class metrics rather than relying solely on overall accuracy.*

Keywords: Data Pipeline; Apache Airflow; Web Scraping; Sentiment Analysis; Ensemble Learning.

Abstrak. Pertumbuhan volume ulasan pengguna pada platform e-commerce di Indonesia berkembang jauh lebih cepat dibandingkan kapasitas analisis konvensional yang bergantung pada peninjauan manual. Penelitian ini merancang dan mengimplementasikan sebuah data pipeline otomatis berbasis arsitektur ETL (Extract, Transform, Load) yang mengintegrasikan akuisisi data melalui web scraping dinamis (Selenium), pra-pemrosesan teks Bahasa Indonesia, klasifikasi sentimen menggunakan model Ensemble Learning (Multinomial Naive Bayes dan Random Forest), orkestrasi alur kerja menggunakan Apache Airflow, kontainerisasi dengan Docker, serta penyimpanan hasil pada basis data PostgreSQL. Sistem diuji menggunakan studi kasus ulasan tiga aplikasi e-commerce populer di Indonesia (Shopee, Tokopedia, dan Blibli) dengan total lebih dari 30.000 baris data ulasan mentah. Hasil pengujian menunjukkan bahwa pipeline berhasil berjalan secara end-to-end dan otomatis, dengan mekanisme idempotensi yang berhasil menjaga integritas data dari risiko duplikasi. Model Ensemble mencapai akurasi keseluruhan sebesar 77,94% dengan teknik SMOTE (Synthetic Minority Over-sampling Technique) untuk menangani ketidakseimbangan kelas. Namun, analisis evaluasi per kelas mengungkapkan bahwa SMOTE hanya efektif memperbaiki performa pada kelas minoritas dengan jumlah data sumber yang relatif memadai, sementara gagal memberikan perbaikan berarti pada kelas dengan ketidakseimbangan ekstrem. Temuan ini memberikan kontribusi metodologis terkait batasan efektivitas teknik oversampling pada kondisi data yang sangat minim, serta menegaskan pentingnya evaluasi metrik per kelas dibandingkan hanya mengandalkan akurasi keseluruhan.

Kata Kunci: Data Pipeline; Apache Airflow; Web Scraping; Analisis Sentimen; Ensemble Learning.

1. LATAR BELAKANG

Ulasan pengguna pada platform aplikasi seperti *Google Play Store* dan ulasan *e-commerce* telah menjadi sumber data yang sangat signifikan bagi pengembang aplikasi untuk memahami kepuasan dan pengalaman pengguna secara langsung (Yasin et al., 2024). Pertumbuhan transaksi *e-commerce* di Indonesia yang meningkat secara konsisten berbanding lurus dengan lonjakan volume ulasan yang dihasilkan setiap harinya, sehingga analisis

konvensional yang bergantung pada peninjauan manual menjadi tidak lagi memadai untuk menangani skala data tersebut (Zakaria & Satyawan, 2023).

Secara tradisional, terdapat dua metode integrasi data utama untuk menangani data berskala besar, yaitu ETL (*Extract, Transform, Load*) dan ELT (*Extract, Load, Transform*). ETL mengekstraksi dan mentransformasikan data di area pemrosesan terpisah sebelum dimuat ke penyimpanan dalam keadaan bersih (Santosh, 2022), sementara ELT memuat data mentah langsung ke data *warehouse* berbasis *cloud* dan mentransformasikannya di dalam *warehouse* tersebut. Pendekatan ETL konvensional umumnya berjalan secara batch dan memerlukan intervensi manual, menimbulkan latensi data dan kerentanan terhadap *human error* (Ogunsola et al., 2022). Di sisi lain, metode analisis sentimen berbasis *machine learning* dalam literatur seringkali hanya berfokus pada optimasi akurasi model, tanpa memperhatikan proses rekayasa data yang menyediakan data secara berkelanjutan dan otomatis (Romadhony et al., 2024). Arsitektur ELT dinilai kurang relevan untuk penelitian ini karena keterbatasan implementasi pada lingkungan lokal (*non-cloud*) dan kebutuhan transformasi menggunakan algoritma *machine learning* kompleks yang lebih efektif dilakukan menggunakan pustaka *Python* sebelum data dimuat ke basis data.

Masih terbatas penelitian yang menyajikan solusi *end-to-end* yang mengintegrasikan akuisisi data, transformasi, pemodelan machine learning, orkestrasi, dan monitoring ke dalam satu arsitektur tunggal, khususnya untuk lingkungan *non-cloud* (Ogunsola et al., 2022). Untuk mengatasi celah tersebut, penelitian ini mengusulkan rancang bangun sebuah data pipeline otomatis dengan arsitektur ETL yang mengintegrasikan *web scraping* dinamis menggunakan *Selenium* (Sharma & Borkar, 2023), pra-pemrosesan teks Bahasa Indonesia, klasifikasi sentimen menggunakan pendekatan *ensemble learning* yang direkomendasikan untuk menangani ketidakseimbangan kelas (George & Srividhya, 2022; Tarawneh et al., 2022), orkestrasi menggunakan *Apache Airflow* (Mbata et al., 2024), containerisasi menggunakan *Docker*, serta penyimpanan hasil pada basis data relasional *PostgreSQL*.

Penelitian ini menjadikan analisis sentimen sebagai studi kasus (*proof-of-concept*) untuk memvalidasi fungsionalitas dan keandalan *pipeline* yang dibangun, bukan sebagai tujuan utama untuk mencapai akurasi model yang maksimal. Tujuan penelitian ini adalah membangun data pipeline yang mampu melakukan proses ETL secara otomatis dari akuisisi data hingga penyimpanan hasil, menerapkan model *ensemble machine learning* sebagai studi kasus untuk membuktikan relevansi data yang dihasilkan *pipeline*, dan mengimplementasikan mekanisme *monitoring* sederhana untuk menyajikan status eksekusi dan kesehatan setiap tugas dalam *pipeline*.

2. KAJIAN TEORITIS

Rekayasa Data dan Arsitektur Data Pipeline

Rekayasa data (*data engineering*) berfokus pada perancangan dan pemeliharaan infrastruktur yang memungkinkan pengumpulan, pemrosesan, dan analisis data secara efisien dan andal (Mbata et al., 2024). Proses ETL merupakan tulang punggung integrasi data dari sistem sumber yang heterogen ke sistem target terpusat untuk keperluan analisis (Ogunsola et al., 2022). Berbeda dari ETL tradisional yang bersifat *batch*, *data pipeline modern* dirancang untuk berjalan secara otomatis, terjadwal, dan dapat dipantau, dengan karakteristik utama berupa otomatisasi, orkestrasi, monitoring dan logging, serta skalabilitas (Ogunsola et al., 2022; Santosh, 2022).

Apache Airflow merupakan platform orkestrasi *open-source* yang merepresentasikan pipeline sebagai *Directed Acyclic Graph* (DAG), yaitu kumpulan tugas yang diatur dalam graf berarah tanpa siklus (Wahyudi et al., 2022). Platform ini dipilih dalam berbagai penelitian rekayasa data karena fleksibilitas pendekatan *pipeline-as-code* dan skalabilitasnya yang teruji di industri, dibandingkan alternatif seperti *AWS Glue* yang cenderung lebih mahal secara operasional pada skala kecil hingga menengah (Mbata et al., 2024; Nayak, 2025).

Web Scraping pada Halaman Dinamis

Tantangan utama dalam akuisisi data dari situs *web modern* adalah konten dinamis yang dimuat secara bertahap menggunakan *JavaScript*, seperti pola *infinite scroll* (Sharma & Borkar, 2023). Pustaka *scraping* statis konvensional seperti *BeautifulSoup* tidak mampu menangani pola ini karena hanya membaca *HTML* awal tanpa menunggu proses *rendering JavaScript* selesai (Muhammad Adhit Dwi Yuda, 2025). *Selenium WebDriver* mengatasi tantangan ini melalui otomatisasi peramban (*browser automation*) yang dapat mensimulasikan interaksi pengguna nyata, termasuk *scrolling*, sehingga seluruh konten dinamis dapat dimuat sepenuhnya sebelum diekstraksi. Studi komparatif menunjukkan bahwa penggunaan *Selenium* secara mandiri dapat mengurangi waktu pemrosesan CPU hingga 70% dan penggunaan data jaringan hingga 90% dibandingkan pendekatan kombinasi lainnya (Sharma & Borkar, 2023).

Analisis Sentimen dan Pra-pemrosesan Teks Bahasa Indonesia

Analisis sentimen merupakan cabang *Natural Language Processing* yang bertujuan menentukan polaritas opini (positif, negatif, atau netral) dalam data teks (Siregar et al., 2025). Alur kerja umumnya melibatkan pra-pemrosesan teks (*cleaning*, *case folding*, normalisasi kata, tokenisasi, *stopword removal*, dan *stemming*), ekstraksi fitur menggunakan TF-IDF, dan klasifikasi menggunakan algoritma *machine learning* (Muhammadi et al., 2022). Untuk teks Bahasa Indonesia yang bersifat informal, normalisasi kata berbasis kamus dan stemming

menggunakan pustaka *Sastrawi* telah terbukti efektif meningkatkan konsistensi data (Romadhony et al., 2024), meskipun penghapusan *stopword* dan *stemming* yang terlalu agresif dapat menghilangkan kata negasi yang krusial bagi polaritas sentimen (Siregar et al., 2025).

Ensemble Learning dan Penanganan Ketidakseimbangan Kelas

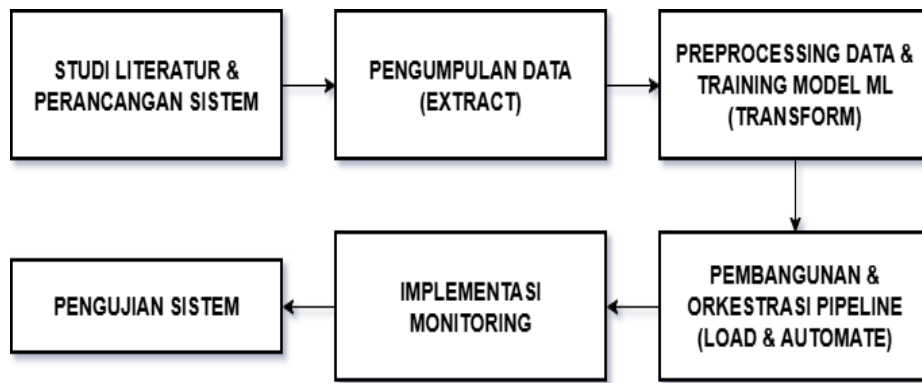
Ensemble Learning menggabungkan prediksi dari beberapa model dasar (*base learners*) untuk menghasilkan keputusan akhir yang lebih akurat dan stabil dibandingkan model tunggal (Agbesi et al., 2024). Pendekatan ini secara khusus direkomendasikan untuk menangani dataset yang tidak seimbang (*imbalanced*), di mana sejumlah tinjauan literatur bahkan menyarankan untuk berhati-hati terhadap teknik *oversampling* seperti *SMOTE* (*Synthetic Minority Over-sampling Technique*) karena risiko menghasilkan data sintesis yang tidak valid, dan merekomendasikan pendekatan ensemble sebagai alternatif yang lebih andal (Tarawneh et al., 2022). Mengingat rekomendasi tersebut, penelitian ini tetap menyertakan *SMOTE* sebagai bagian dari eksperimen pembandingan untuk menguji secara empiris validitas kekhawatiran tersebut pada dataset ulasan *e-commerce* Bahasa Indonesia, dengan hasil yang dibahas secara rinci pada bagian Hasil dan Pembahasan.

3. METODE PENELITIAN

Bagian ini memuat rancangan penelitian meliputi disain penelitian, populasi/ sampel penelitian, teknik dan instrumen pengumpulan data, alat analisis data, dan model penelitian yang digunakan. Metode yang sudah umum tidak perlu dituliskan secara rinci, tetapi cukup merujuk ke referensi acuan (misalnya: rumus uji-F, uji-t, dll). Pengujian validitas dan reliabilitas instrumen penelitian tidak perlu dituliskan secara rinci, tetapi cukup dengan mengungkapkan hasil pengujian dan interpretasinya. Keterangan simbol pada model dituliskan dalam kalimat.

Alur Penelitian

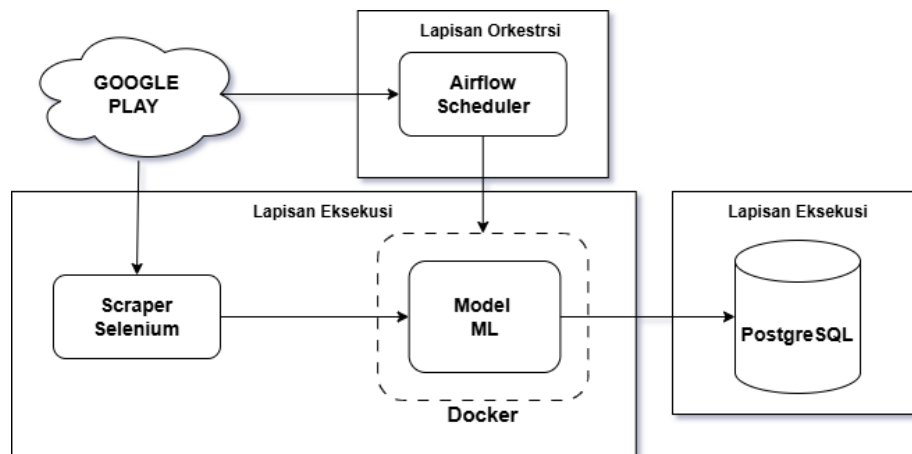
Penelitian ini menerapkan metodologi pengembangan perangkat lunak yang dilakukan secara sekuensial untuk membangun sistem data pipeline. Tahapan penelitian dimulai dari studi literatur dan perancangan sistem, dilanjutkan dengan implementasi teknis yang mencakup pengumpulan data (*extract*), pemrosesan dan transformasi (*transform*), hingga pembangunan dan orkestrasi pipeline (*load & automate*). Tahap akhir melibatkan implementasi mekanisme monitoring dan pengujian sistem secara menyeluruh. Alur kerja penelitian secara keseluruhan diilustrasikan pada Gambar 1.



Gambar 1. Diagram Alur Penelitian.

Arsitektur Sistem

Sistem dirancang menggunakan arsitektur berbasis kontainer (*container-based architecture*) dengan *Docker* untuk menjamin portabilitas dan konsistensi lingkungan kerja. Arsitektur terdiri dari tiga lapisan utama: (1) Lapisan Orkestrasi, ditenagai *Apache Airflow* yang terdiri dari *Webserver* untuk antarmuka pemantauan dan *Scheduler* untuk penjadwalan eksekusi tugas; (2) Lapisan Eksekusi, tempat skrip *Python* menjalankan proses *scraping* (*Selenium*) dan pemrosesan/prediksi sentimen; dan (3) Lapisan Penyimpanan, menggunakan *PostgreSQL* sebagai *metadata store* *Airflow* sekaligus data *warehouse* untuk data mentah, data olahan, dan *log pipeline*.



Gambar 2. Diagram Arsitektur Sistem Data Pipeline Berbasis Docker.

Akuisisi Data

Data dikumpulkan dari ulasan pengguna tiga aplikasi *e-commerce* populer di Indonesia (Shopee, Tokopedia, dan Blibli) pada *Google Play Store*, menggunakan teknik *web scraping* dengan *Selenium WebDriver* yang dijalankan pada mode *headless*. Mengingat ketiadaan API publik untuk pengambilan data ulasan historis secara massal, *scraping* dirancang untuk menangani mekanisme *infinite scroll* melalui simulasi *scrolling* bertahap dengan jeda waktu

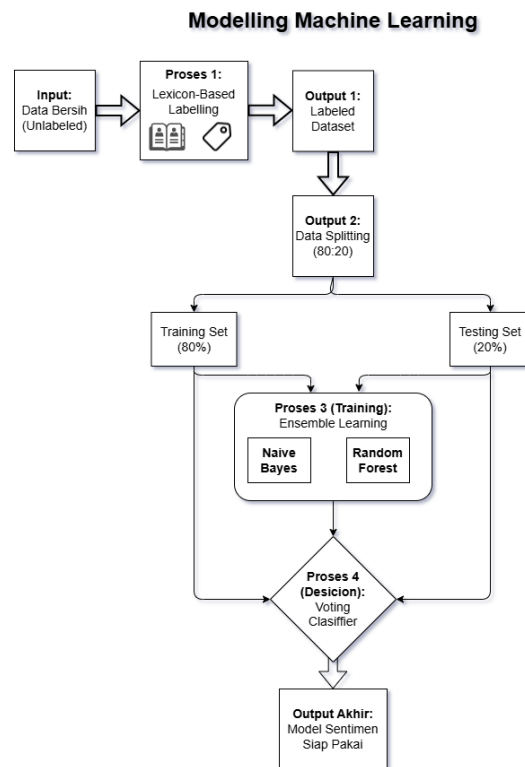
acak (*random delays*) guna meminimalkan risiko terdeteksi sebagai *bot*. Mekanisme penanganan kesalahan (*try-except*) diimplementasikan untuk mengabaikan elemen ulasan yang cacat secara struktur tanpa menghentikan keseluruhan proses ekstraksi. Atribut data yang diekstraksi meliputi nama aplikasi, teks ulasan, tanggal ulasan, dan rating bintang.

Pra-pemrosesan dan Pelabelan Data

Data mentah hasil *scraping* melalui tahapan pra-pemrosesan meliputi pembersihan teks menggunakan Regular Expression (penghapusan tag *HTML*, *URL*, mention, tagar, angka, dan karakter *non-ASCII*), case folding, normalisasi kata berbasis kamus untuk menangani bahasa gaul dan singkatan, tokenisasi, *stopword removal* yang secara selektif mempertahankan kata negasi, dan *stemming* menggunakan pustaka *Sastrawi*. Mengingat tidak memungkinkannya pelabelan manual untuk puluhan ribu data, *ground truth* dibentuk menggunakan pendekatan *lexicon-based labeling* dengan leksikon *InSet (Indonesian Sentiment Lexicon)*, di mana skor sentimen dihitung berdasarkan akumulasi bobot polaritas kata penyusun ulasan: skor total positif dilabeli Positif, skor total negatif dilabeli Negatif, dan skor total nol dilabeli Netral.

Pemodelan Machine Learning

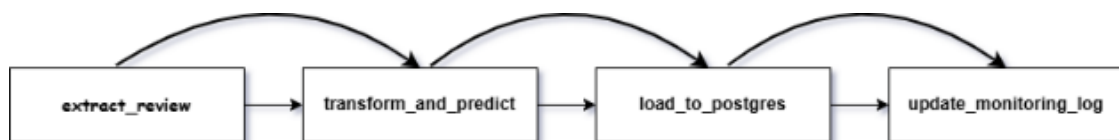
Fitur numerik diekstraksi dari teks bersih menggunakan TF-IDF (*Term Frequency-Inverse Document Frequency*). *Dataset* berlabel dibagi dengan rasio 80:20 secara stratified untuk menjaga proporsi kelas antara data latih dan data uji. Model klasifikasi dibangun menggunakan pendekatan *Ensemble Learning* dengan teknik *Voting Classifier (soft voting)*, menggabungkan dua *base learners*: *Multinomial Naive Bayes*, yang dipilih karena efisiensinya pada klasifikasi teks berbasis frekuensi kata, dan *Random Forest*, yang dipilih karena kemampuannya menangani *noise* dan mengurangi *varians* melalui mekanisme *bagging*. Mengingat distribusi kelas pada data latih yang tidak seimbang, penelitian ini menyertakan eksperimen penerapan *SMOTE* pada fase pelatihan untuk membangkitkan data sintesis pada kelas minoritas, dengan hasil dibandingkan secara langsung terhadap model *baseline* yang dilatih tanpa *SMOTE* sebagaimana disajikan pada bagian Hasil dan Pembahasan.



Gambar 3. Alur Pembangunan Model Machine Learning.

Orkestrasi dan Monitoring

Seluruh alur ETL diintegrasikan sebagai tugas (*task*) dalam satu DAG *Apache Airflow* bernama *sentiment_analysis_pipeline*, dengan urutan dependensi *extract_reviews* >> *transform_and_predict* >> *load_to_postgres* >> *update_monitoring_log*. Notasi >> menunjukkan bahwa tugas di sebelah kanan hanya dijalankan apabila tugas di sebelah kirinya telah selesai dengan status success. Mekanisme idempotensi diterapkan pada proses pemuatan data menggunakan klausa *ON CONFLICT DO NOTHING* pada *PostgreSQL*, berdasarkan batasan unik (*unique constraint*) pada kombinasi nama aplikasi, teks ulasan, dan tanggal ulasan, untuk mencegah duplikasi data pada eksekusi berulang.



Gambar 4. Rancangan DAG (Directed Acyclic Graph) pada *Apache Airflow* untuk Analisis Sentimen.

Pengujian Sistem

Pengujian fungsionalitas sistem dilakukan menggunakan metode *Black Box Testing* untuk memvalidasi alur kerja *end-to-end*, mencakup penjadwalan *Airflow*, ekstraksi data, transformasi data, penyimpanan basis data, mekanisme *monitoring* dan *logging*, serta

penanganan kegagalan. Evaluasi performa model dilakukan menggunakan *Confusion Matrix* dengan empat metrik utama yaitu akurasi, presisi, recall, dan *F1-score*, baik pada level keseluruhan (*weighted average*) maupun pada level masing-masing kelas dan *macro average*, mengingat karakteristik data ulasan yang cenderung tidak seimbang (George & Srividhya, 2022; Romadhony et al., 2024).

4. HASIL DAN PEMBAHASAN

Implementasi Infrastruktur dan Akuisisi Data

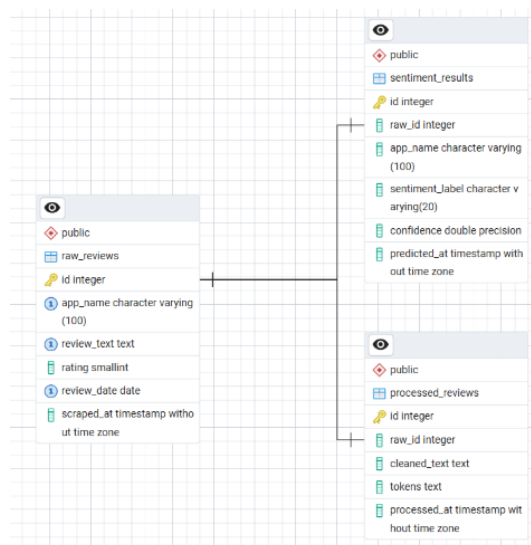
Infrastruktur sistem berhasil dikontainerisasi menggunakan *Docker Compose*, terdiri dari tiga kontainer utama yaitu *airflow_webserver*, *airflow_scheduler*, dan *sentiment_postgres* yang saling terintegrasi dalam satu jaringan virtual. Verifikasi kesehatan kontainer dilakukan menggunakan perintah *docker ps* untuk memastikan seluruh layanan berada pada kondisi *Up* dan *Healthy*

```
(.venv-1) PS Microsoft.PowerShell.Core\FileSystem::\\wsl.localhost\Ubuntu-22.04\home\shandy28\sentiment-pipeline> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
f0e4cd6f0f7b	sentiment-pipeline-airflow-webserver	"/usr/bin/dumb-init ..."	17 seconds ago	Up 9 seconds (health: starting)	0.0.0.0:8081->8080/tcp, [::]:8081->8080/tcp
8c2d90101606	sentiment-pipeline-airflow-init	"/bin/bash -c 'airfl..."	17 seconds ago	Up 9 seconds	8080/tcp
e7e97c6fe193	sentiment-pipeline-airflow-scheduler	"/usr/bin/dumb-init ..."	17 seconds ago	Up 9 seconds (health: starting)	8080/tcp
ab9c7b97f7fc	postgres:15-alpine	"docker-entrypoint.s..."	18 seconds ago	Up 16 seconds (healthy)	0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp

Gambar 5. Hasil eksekusi perintah *docker ps* yang menunjukkan status kontainer *Up* dan *Healthy*.

Basis data *PostgreSQL* diinisialisasi dengan empat tabel relasional: *raw_reviews*, *processed_reviews*, *sentiment_results*, dan *pipeline_logs*, dengan batasan unik pada tabel *raw_reviews* untuk mendukung mekanisme anti-duplikasi.



Gambar 6. Entity-Relationship Diagram basis data *sentiment_db*.

Proses akuisisi data berhasil mengumpulkan lebih dari 30.000 baris data ulasan mentah dari ketiga aplikasi target, dengan atribut meliputi nama aplikasi, teks ulasan, tanggal ulasan, dan rating bintang. Mekanisme mitigasi anti-bot berupa jeda waktu acak dan penanganan kesalahan (*error handling*) untuk elemen ulasan yang rusak struktur *HTML*-nya terbukti efektif menjaga kelangsungan proses scraping tanpa menyebabkan *crash* pada sistem.

```

1 app,text,rating,date,scraped_at
2 shopee,"Dari awal Januari 2026 ini paket saya (terutama yg COD) selalu stuck distatus ""kurir AKAN ditugaskan"" tap
3 shopee,"Izin kasih saran, bagian ulasan produk ketika kasih penilaian beserta foto/video produk mohon bisa tersimpa
4 shopee,"Apk nya bener-bener bagus dan bergunal kebetulan saya pengguna shopee vip jadi banyak sekali voucher yang b
5 shopee,"sedikit masukan buat tim shopee,, 1 = semenjak ada video, pembeli jadi resah, banyak penjual yang menampilkan
6 shopee,"Semakin sering dibuat belanja semakin dipersulit, begitu juga akun toko, baru aja buka toko, ngurus gratis
7 shopee,"Terhusus buat shopee..tolong diperhatikan juga hub daerah masing*,kadang kurir sengaja tidak mengantarkan
8 shopee,"MP paling enak pengiriman cepat walaupun freeongkir ini dulu ya, SEKARANG parah HABIS ekspedisi SPXnya bisa
9 shopee,"Bagus banget, hampir semua barang real pict dan harganya bersaing. Pembatalan, retur barang serta dana juga
10 shopee,"pesanan jadi lambat,dan ya kadang juga kurir sering batalin tanpa konfirmasi katanya ditolak pembeli padaha
11 shopee,"Tolong ya. Perbaiki sistem update lokasi/alamat di aplikasi ini. Saya mengalami kejadian di mana alamat pen
12 shopee,"saya kasih 5 star biar di baca , Tolong dah sistem nya di benarin lagi , terutama di bagian pengiriman untu
13 shopee,"Halo tim Shopee, saya pengguna setia aplikasi ini, namun belakangan saya merasa kurang nyaman dengan sistem
14 shopee,"terima kasih Shopee,dgn adanya shopee saya lebih mudah belanja dan hemat waktu,tapi maaf ya untuk saat saat
15 shopee,"Produk bervariasi dan harga ekonomis tetapi untuk produk yang ukuran besar, ongkirnya bisa 3 - 4 x lipat ha
16 shopee,"aplikasi menyebalkan, lagi browsing kalo muncul iklan shopee pasti langsung pindah aplikasi, iklan yg lain
17 shopee,"Aku pengguna shopee dari tahun 2019 suka dg aplikasi ini kalau buat belanja cuma sekarang kenapa pengirimann
18 shopee,"shopee ekspedisinya sering stuck sekarang berminggu2,,komplain beberapa kali ya masih stuck aja, gak ada peru
19 shopee,"Saya pengguna dari 2014 sampai skrg sudah member platinum tidak ada kelebihan yg bisa di ambil dengan siste
20 shopee,"Saya akui aplikasi Shopee ini bagus buat belanja barang dan kebutuhan, tapi kalau bisa jangan banyak iklann
21 shopee,"Belanja tanpa ragu dengan asuransi. Belanja puas. Tapi aplikasi makan ruang penyimpanan sangat besar hingga
22 shopee,"Fitur chat ai bikin emosi, mau tanya stok atau varian jawabannya muter2 disitu aja, sedangkan saya sering b
23 shopee,"Shopee skrgna pengiriman nya snot lambat. paket saya berminggu"" stuck tidak ad perubahan... Bukan cuma satu

```

Gambar 7. Cuplikan baris data mentah pada berkas `raw_shopee.csv`.

```

[2026-06-20, 09:22:32 UTC] {logger.py:11} INFO - Dec 01, 2021 ChromeDriver version for google-chrome
[2026-06-20, 09:22:32 UTC] {logging_mixin.py:188} WARNING - /home/airflow/.local/lib/python3.12/site-packages/urllib3/connectionpool.py:1103 Inse
[2026-06-20, 09:22:32 UTC] {logger.py:11} INFO - Driver has been saved in cache [/home/airflow/.wdm/drivers/chromedriver/linux64/148.0.7778.178]
[2026-06-20, 09:22:38 UTC] {extractor.py:50} INFO - Mencari tombol 'Lihat semua ulasan'...
[2026-06-20, 09:22:38 UTC] {extractor.py:56} INFO - Tombol berhasil diklik! Kotak ulasan terbuka.
[2026-06-20, 09:22:41 UTC] {extractor.py:61} INFO - Mulai scrolling untuk mencapai target 500 ulasan...
[2026-06-20, 09:22:42 UTC] {extractor.py:69} INFO - Ter-load: 20 ulasan
[2026-06-20, 09:22:47 UTC] {extractor.py:69} INFO - Ter-load: 60 ulasan
[2026-06-20, 09:22:55 UTC] {extractor.py:69} INFO - Ter-load: 100 ulasan
[2026-06-20, 09:23:07 UTC] {extractor.py:69} INFO - Ter-load: 140 ulasan
[2026-06-20, 09:23:14 UTC] {extractor.py:69} INFO - Ter-load: 180 ulasan
[2026-06-20, 09:23:22 UTC] {extractor.py:69} INFO - Ter-load: 220 ulasan
[2026-06-20, 09:23:36 UTC] {extractor.py:69} INFO - Ter-load: 260 ulasan
[2026-06-20, 09:23:43 UTC] {extractor.py:69} INFO - Ter-load: 300 ulasan
[2026-06-20, 09:23:48 UTC] {extractor.py:69} INFO - Ter-load: 340 ulasan
[2026-06-20, 09:23:53 UTC] {extractor.py:69} INFO - Ter-load: 380 ulasan
[2026-06-20, 09:24:15 UTC] {extractor.py:69} INFO - Ter-load: 420 ulasan
[2026-06-20, 09:24:26 UTC] {extractor.py:69} INFO - Ter-load: 460 ulasan
[2026-06-20, 09:24:35 UTC] {extractor.py:69} INFO - Ter-load: 500 ulasan
[2026-06-20, 09:24:35 UTC] {extractor.py:72} INFO - Target 500 tercapai!
[2026-06-20, 09:24:35 UTC] {extractor.py:102} INFO - Mulai mengekstrak data ulasan...
[2026-06-20, 09:25:29 UTC] {extractor.py:153} INFO - Berhasil mengekstrak 500 ulasan dari tokopedia.
[2026-06-20, 09:25:29 UTC] {extractor.py:162} INFO - Browser ditutup.
[2026-06-20, 09:25:29 UTC] {python.py:237} INFO - Done. Returned value was: [{'app_name': 'tokopedia', 'review_text': 'Kupon pengguna baru Tokopec
[2026-06-20, 09:25:29 UTC] {taskinstance.py:441} ► Post task execution logs

```

Gambar 8. Log Airflow yang menunjukkan total statistik ulasan yang berhasil ditarik per eksekusi.

Pra-Pemrosesan Teks

Tahapan pra-pemrosesan berhasil mentransformasi teks ulasan mentah yang mengandung noise (emoji, tanda baca, huruf kapital tidak beraturan, dan kata tidak baku) menjadi teks yang terstandarisasi. Algoritma *stemming* Sastrawi terbukti efektif mereduksi variasi kata berimbuhan ke bentuk dasarnya, misalnya kata “pengiriman”, “dikirim”, dan “mengirim” yang seluruhnya direduksi menjadi “kirim”, sehingga mengurangi dimensi fitur yang harus dipelajari model. Pada beberapa kasus, ditemukan anomali penanganan format tidak

baku, seperti ketiadaan spasi setelah tanda titik pada teks “cancel.pertama” yang menghasilkan keluaran “cancelpertama”, merepresentasikan tantangan nyata dalam menangani data teks dari interaksi kasual pengguna.

Evaluasi Performa Model: Perbandingan Baseline dan SMOTE

Untuk mengukur kontribusi nyata dari teknik penyeimbangan data SMOTE, dilakukan perbandingan antara model Ensemble yang dilatih tanpa penyeimbangan (baseline, menggunakan distribusi label asli hasil *lexicon-based* labeling) dengan model yang dilatih setelah data diseimbangkan menggunakan SMOTE. Populasi awal data latih sebelum SMOTE tercatat sebesar 5.105 sampel Positif, 2.615 sampel Negatif, dan hanya 272 sampel Netral, rasio ketidakseimbangan mendekati 1:19 antara kelas Netral dan Positif.

Tabel 1. Perbandingan Akurasi Keseluruhan Model Klasifikasi: Baseline vs. SMOTE.

Model Klasifikasi	Akurasi (Baseline)	Akurasi (SMOTE)
Naive Bayes	0,7294	0,6508
Random Forest	0,8069	0,7969
Ensemble Model (Final)	0,7779	0,7794

Akurasi keseluruhan model *Ensemble* nyaris tidak berubah antara skenario *baseline* (0,7779) dan *SMOTE* (0,7794). Apabila evaluasi hanya mengandalkan metrik ini, *SMOTE* akan terkesan tidak memberikan pengaruh yang berarti. Namun, evaluasi per kelas pada Tabel 3 mengungkap gambaran yang berbeda secara signifikan.

Tabel 2. Perbandingan Metrik Evaluasi per Kelas: Baseline vs. SMOTE

(n = 1.999 data uji).

Kelas	Skenario	Precision	Recall	F1-Score	Support
Negatif	Baseline	0,85	0,50	0,63	654
	SMOTE	0,68	0,76	0,71	654
Netral	Baseline	0,00	0,00	0,00	68
	SMOTE	0,04	0,01	0,02	68
Positif	Baseline	0,76	0,96	0,85	1.277
	SMOTE	0,85	0,83	0,84	1.277
Macro Avg	Baseline	0,54	0,49	0,49	1.999
	SMOTE	0,52	0,53	0,53	1.999

Pada model baseline, terdapat bias signifikan terhadap kelas Positif, tercermin dari recall Positif yang sangat tinggi (0,96) berbanding recall Negatif yang jauh lebih rendah (0,50) mengindikasikan sekitar separuh ulasan Negatif justru salah diklasifikasikan. Setelah *SMOTE*, *recall* Negatif meningkat menjadi 0,76 dan *F1-score* Negatif meningkat dari 0,63 menjadi 0,71, meski dengan *trade-off* penurunan *precision* (dari 0,85 menjadi 0,68). Pola simetris juga

teramati pada kelas Positif: *recall* menurun (0,96 menjadi 0,83) namun *precision* meningkat (0,76 menjadi 0,85). Hal ini menunjukkan *SMOTE* berhasil mengurangi dominasi kelas Positif dan mendistribusikan keputusan klasifikasi secara lebih proporsional antara dua kelas yang memiliki data sumber memadai (2.615 dan 5.105 sampel).

Pola yang sangat berbeda teramati pada kelas Netral. Model *baseline* mencatat *precision*, *recall*, dan *F1-score* yang bernilai nol secara absolut, model tidak pernah memprediksi satu pun sampel sebagai Netral dari 68 data uji, yang turut menghasilkan peringatan *UndefinedMetricWarning* pada proses evaluasi karena tidak terdapat satu pun prediksi pada label tersebut. Setelah *SMOTE*, *recall* Netral hanya meningkat dari 0,00 menjadi 0,01, secara praktis tidak bermakna secara statistik.

Akar penyebab perbedaan dampak ini berkaitan dengan jumlah data sumber pada masing-masing kelas sebelum *SMOTE* diterapkan. Kelas Netral hanya memiliki 272 sampel pada data latih, jauh di bawah kelas Negatif (2.615) dan Positif (5.105). *SMOTE* membangkitkan data sintetis melalui interpolasi di antara sampel yang sudah ada, dengan basis hanya 272 titik data, interpolasi yang dihasilkan kemungkinan besar terkonsentrasi pada area vektor fitur yang sangat sempit dan tidak merepresentasikan keseluruhan ragam ekspresi linguistik kelas Netral yang sesungguhnya. Akibatnya, peningkatan jumlah sampel secara numerik (dari 272 menjadi 5.105) tidak diiringi peningkatan kemampuan model mengenali kelas tersebut pada data uji.

Temuan ini sejalan dengan kekhawatiran yang dikemukakan oleh Tarawneh et al. (2022) terkait risiko data sintetis yang tidak valid pada teknik *oversampling*, namun memberikan nuansa tambahan *SMOTE* terbukti tetap dapat memberikan kontribusi positif pada kelas minoritas dengan jumlah data sumber yang masih memadai, namun gagal memberikan manfaat berarti pada kelas dengan ketidakseimbangan yang sangat ekstrem. Hal ini mengindikasikan adanya ambang batas (*threshold*) jumlah data sumber minimum bagi efektivitas *SMOTE*, yang juga berimplikasi pada pentingnya melaporkan metrik evaluasi per kelas dan *macro average* dibandingkan hanya mengandalkan akurasi keseluruhan yang dapat menyembunyikan kegagalan pada kelas minoritas ekstrem.

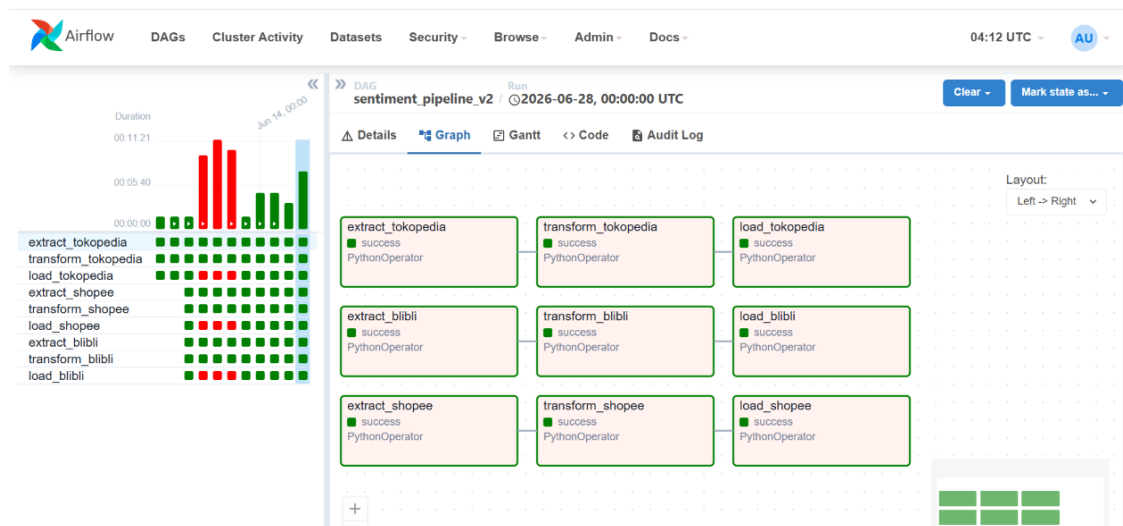
Rendahnya jumlah sampel Netral kemungkinan besar berasal dari karakteristik metode *lexicon-based labeling* itu sendiri, di mana skor sentimen yang persis bernilai nol relatif jarang terjadi pada ulasan yang umumnya mengandung kecenderungan polaritas tertentu, sehingga ulasan dengan sentimen campuran (*mixed sentiment*) cenderung tidak terklasifikasi secara konsisten sebagai Netral.

Orkestrasi dan Monitoring Pipeline

DAG *sentiment_analysis_pipeline* berhasil diimplementasikan dan dijalankan secara terjadwal melalui *Apache Airflow*. Pemantauan melalui fitur *Grid View* menunjukkan riwayat eksekusi yang konsisten berstatus *Success* tanpa mengalami *system crash*, membuktikan keandalan orkestrasi dalam menjalankan rangkaian tugas yang kompleks secara mandiri.



Gambar 9. Visualisasi struktur DAG *sentiment_analysis_pipeline* pada Graph View Airflow.

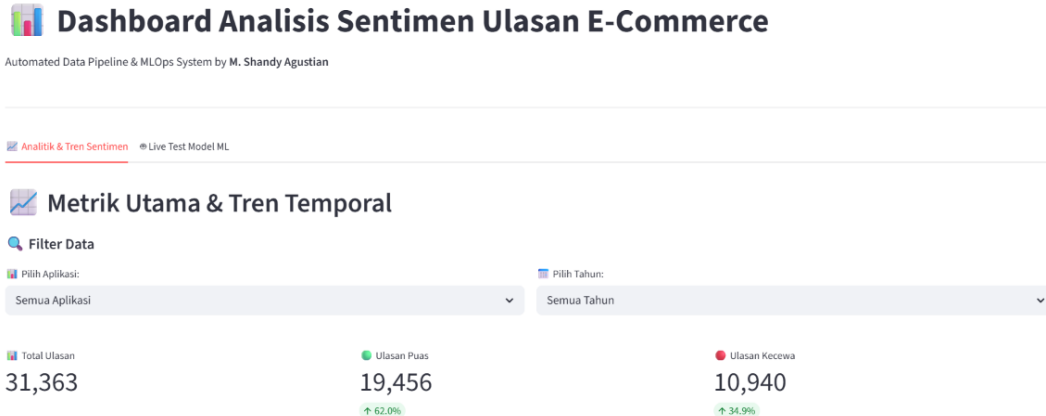


Gambar 10. Riwayat eksekusi DAG yang menunjukkan status Success secara konsisten pada Grid View Airflow.

Mekanisme idempotensi melalui klausa *ON CONFLICT DO NOTHING* pada *PostgreSQL* berhasil menjaga integritas lebih dari 30.000 baris data historis dari risiko duplikasi pada eksekusi berulang.

Visualisasi dan Pengujian Interaktif

Dashboard berbasis Streamlit berhasil menyajikan metrik ringkasan, grafik tren sentimen temporal, distribusi sentimen, dan visualisasi *word cloud* yang terhubung secara *near real-time* dengan basis data melalui mekanisme *caching* bertingkat waktu (TTL 60 detik).



Gambar 11. Tampilan Utama Dashboard Streamlit Dengan Metrik Ringkasan Dan Filter Dinamis.



Gambar 12. Grafik Garis (Line Chart) Fluktuasi Sentimen Per Bulan Dan Bar Chart Distribusi Ulasan Per Aplikasi.



Gambar 13. Word Cloud Kata-Kata Dominan Pada Ulasan Bersentimen Positif (Hijau) Dan Negatif (Merah).

Fitur *Live Sentiment Test* menunjukkan bahwa model tidak hanya berfungsi pada tahap evaluasi statis, namun juga fungsional untuk inferensi dunia nyata. Pengujian interaktif ini turut mengungkap keterbatasan model pada kalimat bersentimen campuran, misalnya kalimat “Aplikasi ini sangat membantu tapi kadang suka lemot” yang diprediksi sebagai Positif. Investigasi menunjukkan bahwa kata tidak baku “lemot” mengalami *over-stemming* oleh Sastrawi menjadi “lot”, sehingga kehilangan polaritas negatifnya, sementara model tidak menangkap efek negasi dari konjungsi “tapi”. Temuan ini menegaskan bahwa penyempurnaan arsitektur NLP yang lebih peka konteks merupakan peluang penelitian lanjutan, sementara fokus utama penelitian ini, keberhasilan otomatisasi aliran data *end-to-end* telah tercapai.

Uji Model Sentimen Langsung (Live Testing)

Ketik ulasan di bawah ini untuk menguji performa model Machine Learning secara langsung.

Masukkan Ulasan Pelanggan:

Aplikasi ini sangat membantu tapi kadang suka lemot

Prediksi Sentimen

Hasil Analisis:

Prediksi Label: POSITIF 😊

Tingkat Keyakinan (Confidence): 66.75%

Teks Preprocessed: aplikasi ini sangat bantu tapi kadang suka lot

Gambar 14. Hasil Pengujian Interaktif Model Pada Kalimat Bersentimen Campuran Melalui Fitur Live Test.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan mengimplementasikan *data pipeline* otomatis *end-to-end* yang mengintegrasikan akuisisi data dinamis, pra-pemrosesan teks Bahasa Indonesia, klasifikasi sentimen berbasis *ensemble learning*, orkestrasi melalui *Apache Airflow*, kontainerisasi *Docker*, dan penyimpanan *PostgreSQL*. Sistem terbukti mampu berjalan secara otomatis tanpa intervensi manual, dengan mekanisme idempotensi yang efektif menjaga integritas lebih dari 30.000 baris data historis.

Model *Ensemble* mencapai akurasi keseluruhan 77,94% setelah penerapan *SMOTE*, namun evaluasi per kelas mengungkap bahwa perbaikan ini tidak merata, *SMOTE* efektif memperbaiki performa pada kelas dengan data sumber memadai (Negatif), namun gagal pada kelas dengan ketidakseimbangan ekstrem (Netral, hanya 272 sampel sumber). Temuan ini memberikan kontribusi metodologis mengenai pentingnya ambang batas data sumber

minimum bagi efektivitas teknik *oversampling*, serta pentingnya pelaporan metrik per kelas dibandingkan hanya akurasi keseluruhan.

Penelitian selanjutnya disarankan untuk mengeksplorasi pengumpulan data tambahan khusus pada kelas minoritas ekstrem, evaluasi ulang ambang batas skor pada metode *lexicon-based labeling*, penggunaan model bahasa berbasis Transformer seperti IndoBERT untuk menangani konteks dan sentimen campuran secara lebih akurat, serta migrasi infrastruktur ke layanan *cloud* terkelola untuk mencapai skalabilitas tingkat produksi.

DAFTAR REFERENSI

- Agbesi, V. K., Chen, W., Yussif, S. B., Ukwuoma, C. C., Gu, Y. H., & Al-Antari, M. A. (2024). MuTCELM: An optimal multi-TextCNN-based ensemble learning for text classification. *Heliyon*, 10(19), e38515. <https://doi.org/10.1016/j.heliyon.2024.e38515>
- George, S., & Srividhya, V. (2022). Performance evaluation of sentiment analysis on balanced and imbalanced dataset using ensemble approach. *Indian Journal of Science and Technology*, 15(17), 790–797. <https://doi.org/10.17485/IJST/v15i17.2339>
- Mbata, A., Sripada, Y., & Zhong, M. (2024). A survey of pipeline tools for data engineering. arXiv. <http://arxiv.org/abs/2406.08335>
- Muhammad Adhit Dwi Yuda. (2025). Implementasi *web scraping* untuk ekstraksi data penjual dan produk panel surya di *e-commerce*. *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, 5(1), 1–12. <https://doi.org/10.24002/konstelasi.v5i1.11751>
- Muhammadi, R. H., Laksana, T. G., & Arifa, A. B. (2022). Combination of support vector machine and lexicon-based algorithm in Twitter sentiment analysis. *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, 8(1), 59–71. <https://doi.org/10.23917/khif.v8i1.15213>
- Nayak, U. (2025). Comparing AWS Glue vs. Apache Airflow for data orchestration: A comprehensive performance and cost analysis. *International Journal of Emerging Trends in Computer Science and Information Technology*, 6(3), 51–55. <https://doi.org/10.63282/3050-9246.IJETCSIT-V6I3P109>
- Ogunsola, K. O., Balogun, E. D., & Ogunmokun, A. S. (2022). Developing an automated ETL pipeline model for enhanced data quality and governance in analytics. *International Journal of Multidisciplinary Research and Growth Evaluation*, 3(1), 791–796. <https://doi.org/10.54660/IJMRGE.2022.3.1.791-796>
- Romadhony, A., Faraby, S. Al, Rismala, R., Wisesti, U. N., & Arifianto, A. (2024). Sentiment analysis on a large Indonesian product review dataset. *Journal of Information Systems Engineering and Business Intelligence*, 10(1), 167–178. <https://doi.org/10.20473/jisebi.10.1.167-178>
- Santosh, K. S. (2022). ETL process automation: Tools and techniques. *ESP Journal of Engineering & Technology Advancements*, 2(1), 1–13. <https://doi.org/10.56472/25832646/JETA-V2I1P110>
- Sharma, K., & Borkar, G. M. (2023). Comparative analysis of dynamic web scraping strategies: Evaluating techniques for enhanced data acquisition. *Advancements in Communication and Systems*, 241–252. <https://doi.org/10.56155/978-81-955020-7-3-22>

- Siregar, M. E., Dermawan, S., & Hisyam, A. A. (2025). Perbandingan kinerja Naive Bayes dan KNN dalam analisis sentimen komentar X dengan dan tanpa *text preprocessing* (Studi kasus: Danantara). *Jurnal Informatika dan Teknik Elektro Terapan*, 13(3). <https://doi.org/10.23960/jitet.v13i3.6612>
- Tarawneh, A. S., Hassanat, A. B., Altarawneh, G. A., & Almuhaimeed, A. (2022). Stop oversampling for class imbalance learning: A review. *IEEE Access*, 10, 47643–47660. <https://doi.org/10.1109/ACCESS.2022.3169512>
- Wahyudi, E. E., Auzan, M., Dharmawan, A., Nuryanto, D. E., Susyanto, N., Samodra, G., & Hadmoko, D. S. (2022). Akuisisi data prediksi curah hujan secara periodik menggunakan Apache Airflow. *Journal of Informatics, Information System, Software Engineering and Applications (INISTA)*, 4(2), 1–12. <https://doi.org/10.20895/inista.v4i2.574>
- Yasin, A., Fatima, R., Ghazi, A. N., & Wei, Z. (2024). Python data odyssey: Mining user feedback from Google Play Store. *Data in Brief*, 54, 110499. <https://doi.org/10.1016/j.dib.2024.110499>
- Zakaria, R., & Satyawati, M. (2023). Strategi implementasi *fintech reward crowdfunding* di Indonesia sektor ekonomi kreatif. *Jurnal Bisnis dan Manajemen West Science*, 2(2), 145–167. <https://doi.org/10.58812/jbmws.v2i02.328>