

Arsitektur Data *Pipeline* Otomatis Berbasis Python untuk *Scraping* X (Twitter): Studi Kasus Program Makan Bergizi Gratis

Kharisma Alya Septiana^{1*}, Arizona Firdonsyah¹

¹Program Studi Teknologi Informasi, Universitas 'Aisyiyah Yogyakarta, Indonesia.

Artikel Info

Kata Kunci:

Data Pipeline;
Pipeline Otomatis;
Program Makan Bergizi Gratis;
Python;
Scraping Twitter (X).

Keywords:

Data Pipeline;
Automated Pipeline;
Free Nutritious Meal Program;
Python;
Twitter (X) Scraping.

Riwayat Artikel:

Submitted: 19 Maret 2026
Accepted: 31 Maret 2026
Published: 31 Maret 2026

Abstrak: Media sosial X (Twitter) merupakan sumber data penting untuk analisis opini publik, namun pembatasan akses API sejak 2023 menghambat pengumpulan data berskala besar secara berkelanjutan. Kondisi tersebut mendorong penggunaan *web scraping* sebagai alternatif. Penelitian ini bertujuan merancang dan mengevaluasi arsitektur *data pipeline* berbasis ETL (*Extract, Transform, Load*) yang terotomatisasi dan terjadwal, dengan studi kasus Program Makan Bergizi Gratis (MBG). *Pipeline* diorkestrasi menggunakan Prefect, melakukan ekstraksi data melalui Tweet Harvest berbasis kata kunci, transformasi melalui pembersihan dan standarisasi data, serta penyimpanan ke basis data relasional MySQL dengan dukungan *monitoring* dan notifikasi otomatis. Hasil menunjukkan *pipeline* berjalan stabil selama tujuh hari pengujian dengan total 8.662 data mentah dan 6.377 data hasil pembersihan. Uji kualitas data menghasilkan nilai *precision* sebesar 95,5%, menunjukkan relevansi tinggi terhadap topik penelitian. Uji performa menunjukkan penggunaan sumber daya rendah dengan rata-rata CPU 0,50% dan RAM 146,34 MB. Dengan demikian, arsitektur *pipeline* ETL otomatis berbasis Python efektif untuk pengumpulan data media sosial secara terstruktur, efisien, dan berkelanjutan sebagai alternatif keterbatasan akses API platform X.

Abstract: Social media platform X (Twitter) is a key data source for public opinion analysis, but API access restrictions implemented in 2023 have hindered the sustainable collection of large-scale data. This situation has led to the use of *web scraping* as an alternative. This study aims to design and evaluate an automated and scheduled ETL (*Extract, Transform, Load*)-based data pipeline architecture, using the Free Nutritious Meals Program (MBG) as a case study. The pipeline is orchestrated using Prefect, performing data extraction via keyword-based Tweet Harvest, transformation through data cleaning and standardization, and storage into a MySQL relational database with support for automated monitoring and notifications. Results show the pipeline ran stably during seven days of testing with a total of 8,662 raw data points and 6,377 cleaned data points. Data quality testing yielded a precision score of 95.5%, indicating high relevance to the research topic. Performance testing showed low resource usage with an average CPU utilization of 0.50% and RAM usage of 146.34 MB. Thus, the Python-based automated ETL pipeline architecture is effective for structured, efficient, and sustainable social media data collection as an alternative to the limited API access of the X platform.

Corresponding Author:

Kharisma Alya Septiana
Email: 2211501004@student.unisayogya.ac.id

PENDAHULUAN

Perkembangan media sosial telah menghasilkan volume data digital yang sangat besar dan terus meningkat, yang mencerminkan dinamika opini publik secara *real-time* terhadap berbagai isu sosial dan kebijakan. Media sosial tidak hanya berfungsi sebagai sarana komunikasi, tetapi juga sebagai ruang ekspresi opini dan penyebaran informasi secara luas (Widcaksono & Silmina, 2023; Suhendra & Pratiwi, 2024; Firdonsyah & Tyas, 2021). Dalam konteks ini, platform X (sebelumnya Twitter) menjadi salah satu sumber utama dalam analisis opini publik karena karakteristiknya yang terbuka dan berbasis *microblog*, sehingga memungkinkan pengumpulan data dalam skala besar (Barutu et al., 2025; Otiva et al., 2024).

Akses terhadap data platform X mengalami keterbatasan signifikan sejak perubahan kebijakan API pada tahun 2023. Pembatasan akses melalui skema berbayar menyebabkan berkurangnya kemampuan peneliti dalam memperoleh data secara berkelanjutan (Murtfeldt et al., 2024). Biaya layanan API tingkat lanjut yang mencapai puluhan ribu USD per bulan menjadikannya tidak terjangkau bagi sebagian besar peneliti akademik (Kupfer, 2024; Blakey, 2024). Kondisi ini mendorong penggunaan *web scraping* sebagai alternatif dalam pengumpulan data publik yang lebih fleksibel (Trezza, 2023).

Di sisi lain, data hasil *scraping* umumnya bersifat tidak terstruktur, mengandung noise, serta memiliki inkonsistensi format yang berpotensi menurunkan kualitas analisis apabila tidak dikelola dengan baik. Kondisi ini menuntut adanya pendekatan sistematis dalam pengelolaan data, salah satunya melalui proses *Extract, Transform, Load* (ETL), yang digunakan untuk memastikan konsistensi, integritas, dan kualitas data sebelum tahap analisis, serta telah terbukti efektif dalam meningkatkan pengelolaan data pada berbagai penelitian sebelumnya (Ogunsola et al., 2022). Pada skala data yang besar dan berkelanjutan, proses ETL perlu dijalankan secara otomatis dan terjadwal guna meningkatkan efisiensi dan konsistensi pemrosesan data (Chanda, 2024).

Penelitian sebelumnya umumnya berfokus pada tahapan tertentu secara terpisah. Beberapa studi memanfaatkan *web scraping* sebagai tahap awal dalam pengumpulan data media sosial yang kemudian digunakan untuk analisis lanjutan, seperti analisis sentimen (Amirullah et al., 2023; Yuniar et al., 2022), sementara penelitian lain berfokus pada implementasi ETL dalam pengelolaan data terstruktur, khususnya pada sistem data warehouse untuk integrasi dan pengolahan data, tanpa melibatkan proses pengambilan data dari sumber tidak terstruktur seperti media sosial (Andriansyah, 2022). Sejumlah studi lain juga telah membangun *pipeline* ETL pada berbagai konteks seperti data akademik, pendidikan, dan transportasi, dengan hasil yang menunjukkan peningkatan efisiensi pengelolaan data, namun masih terbatas pada data terstruktur dan belum didukung mekanisme orkestrasi terjadwal (Ghani & Kurniawan, 2024; Mnsen et al., 2024; Artika et al., 2025).

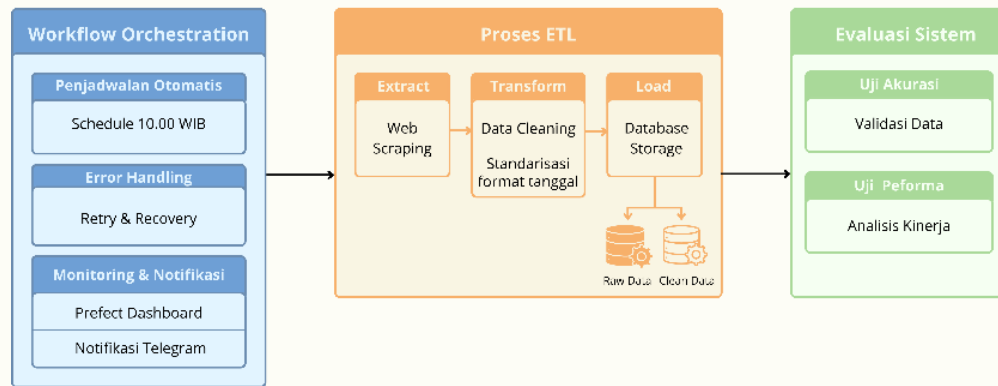
Penelitian lain telah menerapkan ETL pada data media sosial untuk meningkatkan kualitas data hasil *scraping*, namun belum mengintegrasikan proses tersebut dalam *pipeline* yang terotomatisasi dan terorkestrasi. Keterbatasan ini menyebabkan proses pengolahan data menjadi tidak konsisten, sulit dimonitor secara *real-time*, serta rentan terhadap kegagalan eksekusi pada skala data yang besar dan berkelanjutan. Selain itu, proses pengambilan, transformasi, dan penyimpanan data pada konteks data media sosial belum terintegrasi dalam *pipeline* yang terotomatisasi dan terorkestrasi, sehingga menghambat efisiensi serta keberlanjutan pengelolaan data dalam penelitian.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan merancang dan mengevaluasi arsitektur *data pipeline* berbasis ETL dengan dukungan *workflow orchestration* menggunakan Prefect. Penelitian ini mengintegrasikan proses *scraping*, transformasi, dan penyimpanan data dalam satu *pipeline* terorkestrasi yang berjalan secara otomatis dan terjadwal. Selain itu, penelitian ini juga mengevaluasi *pipeline* tidak hanya dari sisi kualitas data hasil *scraping*, tetapi juga dari aspek performa sistem berdasarkan penggunaan sumber daya komputasi.

Implementasi dilakukan pada data diskusi Program Makan Bergizi Gratis (MBG) di platform X sebagai studi kasus karena tingginya intensitas percakapan publik terhadap kebijakan ini serta relevansinya sebagai isu nasional (Zulaika et al., 2025; Kiftiyah et al., 2025).

METODE

Metode penelitian ini berfokus pada pembangunan arsitektur *data pipeline* berbasis ETL (*Extract, Transform, Load*) (Ghani & Kurniawan, 2024), yang terorkestrasi secara otomatis. Arsitektur sistem terdiri atas tiga komponen utama, yaitu *workflow orchestration*, proses ETL (*extract-transform-load*), serta evaluasi sistem. Rancangan keseluruhan sistem ditunjukkan pada Gambar 1.



Gambar 1. Rancangan Sistem ETL Terorkestrasi

Workflow orchestration

Workflow orchestration digunakan untuk mengatur proses penjadwalan, eksekusi, dan pemantauan *pipeline* ETL secara otomatis. Beberapa tools yang umum digunakan antara lain Apache Airflow, Luigi, dan Prefect (Yasmin et al., 2024). Apache Airflow menggunakan pendekatan Directed Acyclic Graph (DAG) untuk mengelola workflow dengan komponen seperti scheduler dan metadata database, sedangkan Luigi mengelola workflow berbasis dependency antar task dengan pendekatan yang lebih sederhana.

Pipeline dalam penelitian ini memiliki karakteristik berupa proses *scraping* yang rentan terhadap kegagalan, membutuhkan mekanisme *retry*, serta kemungkinan keterlambatan eksekusi akibat downtime server. Pada kondisi tersebut, pendekatan DAG statis pada Airflow menjadi kurang fleksibel karena dependency harus ditentukan di awal dan sulit beradaptasi terhadap kondisi saat runtime. Sementara itu, Luigi memiliki keterbatasan pada fitur *monitoring* dan observability, sehingga kurang optimal untuk pemantauan *pipeline* secara *real-time*. Berdasarkan kebutuhan tersebut, Prefect dipilih karena mendukung *dynamic workflow* yang memungkinkan penyesuaian task saat runtime, sehingga sesuai dengan karakteristik proses *scraping* yang tidak deterministik. Selain itu, Prefect menyediakan fitur *retry*, pencatatan *state task*, serta mekanisme *late run recovery* yang digunakan dalam penelitian ini untuk menangani kegagalan eksekusi akibat gangguan jaringan atau downtime server. Fitur *monitoring* terintegrasi juga memungkinkan pemantauan *pipeline* secara *real-time* melalui dashboard (Baiochi et al., 2025; Staneva et al., 2024).

Prefect mengatur eksekusi flow yang terdiri dari beberapa task utama, yaitu proses ekstraksi data, transformasi data, dan pemuatan data ke dalam basis data. *Pipeline* ETL di-deploy pada server berbasis Ubuntu 22.04 dengan lingkungan Python 3.12, di mana script dijalankan langsung sehingga *worker* Prefect dapat mengeksekusi flow sesuai jadwal yang telah ditentukan. Dalam implementasinya, Prefect digunakan melalui layanan cloud versi gratis yang memiliki keterbatasan dalam penyimpanan riwayat eksekusi workflow. Namun, keterbatasan ini tidak mempengaruhi tujuan penelitian karena *monitoring* difokuskan pada keberhasilan eksekusi *pipeline* selama proses pengujian berlangsung.

1. Penjadwalan Otomatis

Penjadwalan otomatis pada *pipeline* ETL dilakukan untuk memastikan eksekusi *flow* berjalan secara teratur sesuai konfigurasi yang telah ditetapkan (Chanda, 2024). Pengaturan jadwal menggunakan format *cron* 0 10 * * *, yang berarti *pipeline* dijalankan satu kali setiap hari pada pukul 10.00 WIB. Pada waktu yang telah dijadwalkan tersebut, sistem secara otomatis mengeksekusi tahapan

ETL secara berurutan, dimulai dari proses ekstraksi data, dilanjutkan dengan transformasi, hingga pemuatan data ke dalam basis data. *Pipeline* memproses data pada rentang waktu satu hari sebelumnya (00.00–23.59 WIB) untuk memastikan kelengkapan data dalam satu siklus *batch* harian.

2. Error Handling

Pada *workflow orchestration*, penanganan error untuk skenario *downtime* server diterapkan dengan menjalankan *worker* sebagai *startup service* menggunakan *trigger AtStartup* (tanpa ketergantungan login pengguna). Dengan konfigurasi ini, ketika server kembali hidup setelah jadwal terlewat, *worker* otomatis aktif dan sistem memproses *run* tertunda (*late run recovery*) secara otomatis. Pendekatan ini memastikan keberlangsungan eksekusi ETL tetap terjaga meskipun terjadi gangguan operasional sementara.

Pada kondisi server mati saat proses sedang berjalan, *run* aktif pada Prefect dapat berstatus *Crashed*. Namun, kondisi ini tidak menghentikan *pipeline* secara permanen karena eksekusi berikutnya tetap dijalankan ketika *worker* kembali aktif. Pada tahap ekstraksi, sistem mencatat keyword terakhir yang telah berhasil diproses sehingga ketika terjadi gangguan eksekusi, proses ekstraksi dapat dilanjutkan dari keyword berikutnya tanpa mengulang seluruh proses *scraping*. Dengan demikian, mekanisme *error handling* tidak hanya menangani keterlambatan jadwal, tetapi juga memungkinkan eksekusi yang terhenti untuk dilanjutkan kembali.

3. Monitoring dan Notifikasi

Monitoring dilakukan untuk memantau status eksekusi *pipeline* ETL yang dijalankan secara terjadwal melalui *dashboard* visual bawaan Prefect yang memberikan pandangan komprehensif tentang proses ETL sehingga memudahkan pemantauan dan pengelolaan alur kerja (Baiochi et al., 2025). Informasi waktu mulai, waktu selesai, durasi, serta status akhir setiap eksekusi tercatat secara otomatis dalam sistem orkestrasi.

Sebagai pendukung *monitoring*, notifikasi dikirimkan secara otomatis melalui Telegram ketika proses dimulai, selesai, maupun saat terjadi kegagalan. Telegram dipilih karena kemudahan aksesnya sebagai platform *monitoring* yang responsif, menyediakan *Bot API* untuk pengiriman notifikasi secara *real-time*, serta merupakan solusi yang hemat biaya dengan kemampuan integrasi yang fleksibel terhadap berbagai sistem (Bintang et al., 2025). Mekanisme ini memungkinkan pengawasan *pipeline* dilakukan secara responsif tanpa perlu pemeriksaan langsung pada lingkungan eksekusi.

Proses ETL (*Extract, Transform, Load*)

Proses ETL (*Extract, Transform, Load*) dilaksanakan untuk mengelola data mulai dari tahap pengambilan, pembersihan, hingga penyimpanan ke dalam basis data relasional.

1. Extract (Pengambilan Data)

Tahap ekstraksi dilakukan dengan mengambil data publik dari platform X (Twitter) menggunakan metode *web scraping* berbasis kata kunci sebagai alternatif terhadap keterbatasan akses API (Fadhlullah et al., 2022). Proses *scraping* dilakukan menggunakan Tweet Harvest (Lediwara et al., 2024), sebuah *tool* berbasis *command-line* yang memungkinkan pengambilan *tweet* berdasarkan kata kunci dan rentang waktu tertentu. Ekstraksi menggunakan tujuh kata kunci yang merepresentasikan variasi istilah terkait Program Makan Bergizi Gratis (MBG), yaitu “makan bergizi gratis”, “program makan bergizi”, “MBG”, “makan gratis sekolah”, “gizi anak sekolah”, “program gizi pemerintah”, dan “bantuan makan gratis”. Variasi ini digunakan untuk memperluas cakupan data dan menangkap beragam bentuk penyebutan dalam percakapan publik. Pendekatan berbasis kata kunci memiliki keterbatasan dalam menjangkau seluruh percakapan yang relevan (*low recall*). Untuk mengurangi dampak tersebut, kualitas hasil ekstraksi dievaluasi menggunakan metrik *precision* melalui validasi manual terhadap sampel data untuk memastikan kesesuaian antara hasil ekstraksi dan konteks yang diinginkan.

Pengambilan data dilakukan pada periode Februari–Maret 2026 dengan filter bahasa Indonesia (lang:in). Parameter query mencakup kata kunci, rentang tanggal, serta batas jumlah data per eksekusi. Setiap eksekusi dibatasi maksimum 500 tweet per kata kunci dalam satu siklus *batch* harian. Batas ini

merupakan nilai maksimum, sehingga jumlah data yang diperoleh dapat bervariasi tergantung ketersediaan tweet. Batas tersebut diterapkan untuk menjaga stabilitas proses *scraping*, menghindari pembatasan akses dari platform, serta memastikan proses ekstraksi dapat berjalan secara konsisten dalam setiap siklus pengambilan data. Data hasil ekstraksi meliputi atribut `id_str`, `full_text`, `created_at`, `lang`, `username`, `location`, `source_file`, `like_count`, `retweet_count`, dan `reply_count`. Seluruh data yang diperoleh disimpan terlebih dahulu ke dalam tabel `raw_data` guna mempertahankan integritas data sumber sebelum memasuki tahap transformasi.

Mengingat metode *scraping* rentan terhadap pembatasan akses, gangguan koneksi jaringan, serta perubahan struktur halaman, tahap ekstraksi dilengkapi dengan mekanisme mitigasi berupa penerapan jeda *scraping* selama 20–40 detik antar proses serta mekanisme *retry* hingga maksimal tiga kali dengan jeda bertahap 60, 90, dan 120 detik. Perubahan struktur halaman dapat memengaruhi keberhasilan proses *scraping*, sehingga diperlukan penyesuaian mekanisme ekstraksi apabila terjadi perubahan signifikan. Pendekatan yang lebih adaptif, seperti penggunaan teknik berbasis kecerdasan buatan, telah diusulkan dalam penelitian sebelumnya untuk mengatasi dinamika tersebut, namun tidak menjadi fokus dalam penelitian ini.

Selain aspek teknis, proses pengambilan data juga mempertimbangkan aspek etika serta kepatuhan terhadap Terms of Service (ToS) platform X. Platform media sosial memiliki kebijakan ToS yang mengatur akses dan penggunaan data, yang mencerminkan kontrak sosial antara platform dan publik serta menunjukkan bahwa kendali atas akses data berada pada pihak platform (Blakey, 2024). Dalam konteks penggunaan *web scraping*, pengumpulan data otomatis dapat menimbulkan implikasi etis serta potensi pelanggaran terhadap ketentuan penggunaan, termasuk risiko terhadap subjek maupun platform, sehingga perlu dilakukan secara hati-hati dan bertanggung jawab (Brewer et al., 2021). Oleh karena itu, data yang dikumpulkan dalam penelitian ini hanya berasal dari konten publik tanpa mengakses akun privat atau melewati mekanisme autentikasi, serta tidak mencakup informasi sensitif pengguna. Selain itu, proses *scraping* dilakukan dengan pembatasan volume permintaan dan pemberian jeda akses untuk menghindari beban berlebih pada sistem, sejalan dengan prinsip praktik *web scraping* yang tidak mengganggu layanan platform (Flores et al., 2020).

2. Transform (Pemrosesan Data)

Transformasi dilakukan untuk membersihkan dan menyeragamkan data sebelum penyimpanan (Amanda & Waspodo, 2025; Kridono & Maukar, 2025). Proses pembersihan teks (*data cleaning*) ini dilakukan secara terprogram menggunakan *library* Pandas untuk manipulasi tabel dan *library* *re* (*Regular Expression*) dalam Python untuk melakukan penghapusan pola URL, *tag*, serta karakter non-alfanumerik.

Dari seluruh atribut hasil ekstraksi, hanya kolom `id_str`, `full_text`, `created_at`, `lang`, dan `source_file` yang dipertahankan karena secara langsung terlibat dalam proses transformasi dan pembentukan struktur data akhir. Atribut `id_str` digunakan untuk identifikasi unik dan penghapusan duplikasi, `full_text` sebagai objek utama pembersihan teks, `created_at` untuk proses standarisasi waktu, serta `lang` dan `source_file` sebagai informasi pendukung dalam pengelolaan dan pelacakan data. Sementara itu, atribut seperti `username` dan `location` tidak disertakan karena tidak tersedia pada hasil ekstraksi serta tidak berperan dalam proses transformasi. Atribut metrik interaksi seperti `like_count`, `retweet_count`, dan `reply_count` juga tidak dipertahankan karena tidak mengalami proses transformasi maupun memiliki fungsi dalam tahapan pembersihan dan standarisasi data, sehingga penyertaannya tidak memberikan kontribusi terhadap hasil akhir *pipeline*. Tahapan transformasi meliputi:

- a. Data Cleaning: Proses pembersihan data yang meliputi penghapusan data duplikat, penanganan *missing value*, penghapusan *tag*, URL, *emoji*, dan tanda baca non-informasi serta *case folding* yaitu penyeragaman seluruh huruf dalam teks menjadi huruf kecil (*lower case*) untuk menghindari perbedaan representasi kata akibat variasi huruf besar dan kecil (Maskhuri & Hardiani, 2025).
- b. Standarisasi format tanggal: Atribut `created_at` pada data mentah yang semula berbentuk *string* (misalnya *Sun Feb 15 23:57:37 +0000 2026*) dikonversi menjadi tipe data *datetime* dengan format *YYYY-MM-DD HH:MM:SS* serta disesuaikan ke zona waktu WIB (UTC+7) sebelum disimpan pada tabel `clean data`.

Untuk memastikan kualitas hasil transformasi, khususnya pada tahap *data cleaning*, dilakukan evaluasi secara kuantitatif menggunakan metrik *precision* melalui validasi manual terhadap sampel data. Evaluasi ini bertujuan mengukur tingkat relevansi data setelah proses pembersihan serta menilai efektivitas transformasi dalam mengurangi *noise*. Pada penelitian ini, proses *stemming* dan *tokenisasi* tidak diterapkan karena *pipeline* difokuskan pada penyediaan data bersih dan terstruktur sebagai *input* untuk analisis lanjutan, bukan pada tahap pemodelan teks. Selain itu, penggunaan teks utuh (*full text*) dipertahankan untuk menjaga konteks asli percakapan dan menghindari potensi kehilangan makna akibat reduksi kata.

3. Load (Penyimpanan Data)

Tahap *load* merupakan bagian akhir dari proses ETL yang berfungsi untuk memuat data hasil transformasi ke dalam sistem penyimpanan secara terstruktur (Artika et al., 2025). Pada penelitian ini, data disimpan ke dalam basis data relasional menggunakan MySQL sebagai media penyimpanan. Pemilihan MySQL didasarkan pada perannya sebagai salah satu sistem manajemen basis data relasional yang banyak digunakan dalam pengelolaan data terstruktur di lingkungan akademik dan penelitian, serta kemampuannya mendukung standar SQL dan integrasi dengan aplikasi berbasis Python.

Data disimpan dalam dua tabel terpisah, yaitu *raw_data* dan *clean_data*. Tabel *raw_data* menyimpan hasil ekstraksi awal tanpa perubahan, sedangkan tabel *clean_data* menyimpan data yang telah melalui proses pembersihan dan standarisasi. Pemisahan ini bertujuan menjaga integritas data asli serta memudahkan proses penelusuran apabila terjadi perbedaan antara data sebelum dan sesudah transformasi. Atribut *id_str* digunakan sebagai *unique key* untuk mencegah penyimpanan data yang sama lebih dari satu kali. Setiap siklus eksekusi harian menambahkan data baru ke dalam tabel tanpa menghapus data yang telah tersimpan sebelumnya.

Evaluasi Sistem

Evaluasi sistem dilakukan untuk menilai kinerja arsitektur *data pipeline* yang telah diimplementasikan. Evaluasi difokuskan pada dua aspek utama, yaitu kualitas data hasil *scraping* dan efisiensi teknis selama proses eksekusi otomatis dan terjadwal.

1. Uji Akurasi Hasil Scraping

Uji akurasi dilakukan untuk menilai kualitas data hasil *scraping* berdasarkan tingkat relevansi data yang diperoleh terhadap kebutuhan penelitian (Javier et al., 2025). Evaluasi menggunakan metrik *precision*, yaitu rasio jumlah data relevan yang berhasil diperoleh terhadap total data yang dianalisis. *Precision* mencerminkan tingkat ketepatan sistem dalam menghasilkan data yang sesuai dengan tujuan pengambilan informasi, sehingga nilai *precision* yang tinggi menunjukkan bahwa sebagian besar data yang diperoleh relevan dengan kebutuhan penelitian. Evaluasi dilakukan terhadap sampel data yang dipilih secara acak menggunakan metode *simple random sampling*. Sebanyak 200 *tweet* digunakan sebagai sampel yang dianggap representatif terhadap populasi, dengan mempertimbangkan keterbatasan proses validasi manual. Pada penelitian ini, data hasil *scraping* diklasifikasikan menjadi data relevan dan tidak relevan berdasarkan kesesuaiannya dengan topik yang diteliti. Selanjutnya, nilai *precision* dihitung dengan membandingkan jumlah data relevan terhadap total data yang berhasil dikumpulkan oleh sistem, sebagaimana dirumuskan pada Persamaan (1).

$$Precision = \frac{Tweet\ Relevan}{Total\ Tweet\ Sampel} \times 100\% \quad (1)$$

Precision dipilih karena total populasi *tweet* relevan pada platform X tidak dapat diketahui secara pasti, sehingga metrik *recall* tidak dapat dihitung secara valid dalam konteks *scraping* data publik.

2. Uji Performa Sistem

Uji performa dilakukan untuk mengevaluasi efisiensi teknis *pipeline* ETL berdasarkan penggunaan CPU (%), RAM (MB), dan durasi eksekusi (menit). Pengukuran dilakukan menggunakan library *psutil* pada Python untuk mencatat penggunaan CPU dan RAM selama eksekusi *pipeline*. Nilai

rata-rata penggunaan CPU, RAM, dan durasi total eksekusi dihitung pada setiap siklus *batch* harian selama periode pengujian tujuh hari. Parameter evaluasi meliputi:

1. Rata-rata penggunaan CPU (%).
2. Rata-rata penggunaan RAM (MB).
3. Durasi total eksekusi per siklus (menit).

HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil pengujian arsitektur *data pipeline* ETL serta pembahasan kinerja sistem berdasarkan aspek penjadwalan, kualitas data, dan efisiensi teknis.

Implementasi *Workflow orchestration*

Bagian ini menguraikan implementasi *workflow orchestration* dalam mengatur dan memantau eksekusi *pipeline* ETL secara otomatis.

1. Penjadwalan Otomatis

Implementasi penjadwalan otomatis menunjukkan bahwa *pipeline* ETL berjalan sesuai konfigurasi *cron* 0 10 * * *, yaitu satu kali setiap hari pukul 10.00 WIB. Selama periode pengujian tujuh hari, seluruh eksekusi terpicu secara otomatis dan memproses data rentang H-1 (00.00–23.59 WIB) secara konsisten. Ringkasan hasil eksekusi harian ditunjukkan pada Tabel 1.

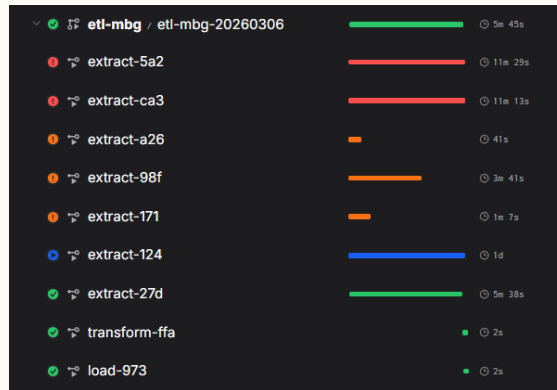
Tabel 1. Ringkasan Hasil Eksekusi Harian *Pipeline* ETL Selama Periode Pengujian

Hari Ke-	Waktu Mulai	Waktu Selesai	Durasi	Status
1	10:00:04	10:34:46	34 menit 42 detik	Completed
2	09:59:59	10:32:20	32 menit 21 detik	Completed
3	10:00:01	10:34:41	34 menit 40 detik	Completed
4	10:00:02	10:56:20	56 menit 20 detik	Completed
5	10:00:00	10:36:58	36 menit 58 detik	Completed
6	10:00:00	10:33:11	33 menit 11 detik	Completed
7	10:00:00	10:29:28	29 menit 28 detik	Completed

Berdasarkan Tabel 1, waktu mulai eksekusi berada pada rentang 09:59:59 hingga 10:00:04. Selisih waktu dalam hitungan detik tersebut menunjukkan adanya variasi teknis yang sangat kecil pada proses pemicu sistem, namun seluruh eksekusi tetap berlangsung pada waktu yang sesuai dengan jadwal pukul 10.00 WIB. Durasi eksekusi *pipeline* berada pada kisaran 32 hingga 35 menit pada setiap pengujian. Perbedaan durasi antar hari menunjukkan adanya variasi beban pemrosesan selama proses ekstraksi dan transformasi data. Meskipun demikian, seluruh proses berhasil diselesaikan dengan status Completed, yang menunjukkan bahwa sistem mampu menjalankan proses secara otomatis dan konsisten selama periode pengujian.

2. Error Handling

Selama periode pengujian selama tujuh hari, *pipeline* ETL berhasil dijalankan sesuai jadwal tanpa ditemukan kegagalan eksekusi. Seluruh *run* berhasil diselesaikan dan dilanjutkan ke tahap berikutnya sesuai dengan alur *pipeline*. Namun, pada pengoperasian sistem setelah periode pengujian, pernah ditemukan kondisi di mana *run* pada tahap ekstraksi berstatus *Crashed*. Meskipun demikian, kondisi tersebut tidak menghentikan *pipeline* secara permanen karena *workflow orchestration* tetap menjalankan *run* berikutnya ketika *worker* kembali aktif. Kondisi ini dapat dilihat seperti yang ditunjukkan pada Gambar 2, di mana salah satu *run* ekstraksi mengalami kegagalan, tetapi *pipeline* tetap melanjutkan eksekusi pada *run* berikutnya.



Gambar 2. Tampilan *flow run* Prefect dengan status *Crashed* pada tahap ekstraksi.

Gambar 2 menunjukkan bahwa pada salah satu siklus eksekusi, beberapa *task extract* berstatus *Crashed* sehingga *run* tidak dapat diselesaikan pada percobaan tersebut. Namun, kegagalan ini tidak menghentikan *pipeline* secara permanen karena *workflow orchestration* secara otomatis menjadwalkan ulang *run* yang tertunda. Pada beberapa kondisi, proses pemulihan tidak selalu berhasil dalam satu percobaan sehingga *run* dapat kembali gagal sebelum akhirnya berhasil dijalankan. Setelah *worker* kembali stabil, *pipeline* melanjutkan eksekusi hingga tahap *transform* dan *load*. Kondisi ini menunjukkan bahwa mekanisme *error handling* mendukung pemulihan otomatis (*late run recovery*) ketika terjadi gangguan operasional sementara.

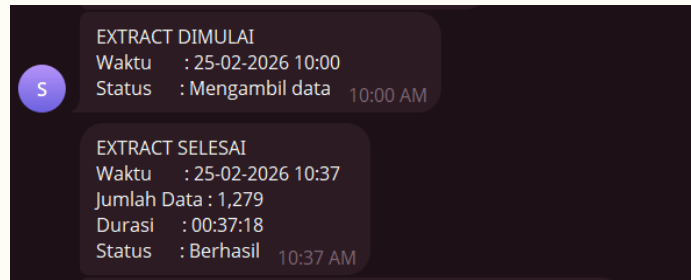
3. Hasil Monitoring dan Notifikasi

Monitoring dilakukan melalui sistem *workflow orchestration* yang mencatat setiap eksekusi *pipeline* secara otomatis, termasuk waktu mulai, waktu selesai, durasi, serta status akhir proses. Selama periode pengujian tujuh hari, seluruh *run* tercatat dengan status *Completed*. Ringkasan tampilan *monitoring* ditunjukkan pada Gambar 3.



Gambar 3. Dashboard Monitoring Pipeline ETL pada Prefect

Pada sistem *monitoring*, status eksekusi divisualisasikan menggunakan indikator warna, yaitu *Completed* (hijau), *Running* (biru), *Late* (kuning), *Failed* (merah), dan *Canceled* (abu-abu). Setiap indikator digunakan untuk menunjukkan kondisi eksekusi *pipeline* secara visual pada *dashboard*. Selama periode pengujian, seluruh *run* berada pada status *Completed*, yang menunjukkan bahwa *pipeline* berjalan sesuai jadwal tanpa mengalami keterlambatan maupun kegagalan proses. Sebagai pendukung *monitoring*, sistem mengirimkan notifikasi otomatis melalui Telegram pada saat proses dimulai dan setelah proses selesai. Contoh notifikasi ditunjukkan pada Gambar 4.

Gambar 4. Contoh Notifikasi *Pipeline* di Telegram

Notifikasi yang dikirimkan memuat informasi waktu eksekusi, jumlah data yang diperoleh, durasi pemrosesan, serta status akhir proses. Mekanisme ini memungkinkan pemantauan dilakukan secara langsung tanpa perlu mengakses *dashboard* sistem. Integrasi tersebut menunjukkan bahwa *pipeline* berjalan otomatis dan tetap terpantau secara konsisten selama periode pengujian.

Proses ETL (*Extract, Transform, Load*)

Bagian ini menyajikan hasil implementasi proses Extract, Transform, dan Load (ETL) yang dijalankan secara otomatis dan terjadwal selama periode pengujian.

1. Extract (Pengambilan Data)

Selama periode pengujian tujuh hari, proses ekstraksi berhasil mengumpulkan data secara otomatis pada setiap siklus batch harian. Total data yang diperoleh selama periode tersebut mencapai 8.665 entri, yang langsung disimpan pada tabel *raw_data* tanpa perubahan untuk menjaga integritas data sumber sebelum tahap transformasi. Rincian jumlah data hasil extract ditunjukkan pada Tabel 2.

Tabel 2. Jumlah Data Hasil Extract Selama Periode Pengujian

Hari Ke-	Jumlah Data
1	1.738
2	1.210
3	1.305
4	1.103
5	1.265
6	1.057
7	984
Total	8.662

Berdasarkan data pada Tabel 2, jumlah tweet yang diperoleh setiap hari mengalami perubahan, dengan jumlah tertinggi sebesar 1.738 tweet dan terendah sebesar 984 tweet. Hal ini menunjukkan bahwa intensitas pembahasan terkait Program Makan Bergizi Gratis (MBG) tidak selalu sama setiap hari. Meskipun seluruh data yang diperoleh berkaitan dengan topik MBG, tingkat perhatian publik terhadap isu tersebut dapat meningkat atau menurun seiring perkembangan situasi. Perubahan jumlah tweet tersebut kemungkinan dipengaruhi oleh berbagai faktor, seperti pemberitaan di media, pernyataan tokoh publik, atau topik yang sedang ramai dibahas di platform X. Dengan demikian, jumlah data yang diperoleh mencerminkan dinamika pembahasan yang terjadi di masyarakat dan perlu diperhatikan dalam analisis lanjutan.

Selain variasi jumlah data per hari, distribusi tweet berdasarkan kata kunci juga menunjukkan adanya ketidakseimbangan kontribusi data. Kata kunci yang bersifat umum dan sering digunakan, seperti “makan bergizi gratis” dan “MBG”, cenderung menghasilkan jumlah data yang lebih banyak dibandingkan kata kunci yang lebih spesifik seperti “gizi anak sekolah” atau “program gizi pemerintah”. Kondisi ini menunjukkan bahwa penggunaan kata kunci tertentu dapat mendominasi data yang diperoleh. Akibatnya, data yang terkumpul lebih banyak merepresentasikan istilah yang populer di masyarakat dibandingkan keseluruhan aspek kebijakan secara merata. Meskipun

penggunaan variasi kata kunci bertujuan untuk memperluas cakupan data, pendekatan berbasis keyword tetap memiliki keterbatasan dalam menghasilkan distribusi data yang seimbang.

Selama periode pengujian, tidak ditemukan kegagalan maupun penghentian proses pada tahap extract. Seluruh run berhasil diselesaikan dan dilanjutkan ke tahap transformasi. Hal ini menunjukkan bahwa mekanisme mitigasi pada proses ekstraksi, seperti pembatasan jumlah data per eksekusi, jeda *scrapping*, serta mekanisme *retry*, mampu menjaga proses pengambilan data tetap berjalan secara konsisten. Secara keseluruhan, tahap extract berhasil menyediakan data mentah secara berkelanjutan untuk mendukung tahap *pipeline* ETL berikutnya.

2. Transform (Pemrosesan Data)

Selama periode pengujian tujuh hari, tahap transform menghasilkan perubahan jumlah data dibandingkan dengan data mentah hasil ekstraksi. Dari total 8.662 entri pada tabel *raw_data*, jumlah data yang tersimpan pada tabel *clean_data* setelah proses transform menjadi 6.377 entri. Rincian perbandingan ditunjukkan pada Tabel 3.

Tabel 3. Perbandingan Jumlah Data Sebelum dan Sesudah Proses Transformasi

Tahap	Jumlah Data
Raw Data	8.662
Clean Data	6.377
Data Tereliminasi	2.285

Sebanyak 6.377 entri (73,62%) dipertahankan sebagai *clean data*, sedangkan 2.285 entri (26,38%) dieliminasi selama proses transformasi. Eliminasi data dilakukan melalui beberapa tahapan pembersihan, yaitu penghapusan data duplikat berdasarkan atribut *id_str*, penghapusan entri dengan nilai kosong pada atribut penting (*id_str*, *created_at*, dan *full_text*), serta proses pembersihan teks untuk memastikan relevansi konten. Proporsi eliminasi ini menunjukkan peran tahap transform dalam menyaring data yang tidak memenuhi kriteria penelitian. Selain perubahan jumlah data, tahap transform juga menghasilkan perubahan pada struktur atribut waktu. Atribut *created_at* yang semula berbentuk string dikonversi menjadi tipe datetime dengan format standar YYYY-MM-DD HH:MM:SS dan disesuaikan dengan zona waktu WIB (UTC+7). Proses ini bertujuan untuk memastikan konsistensi format waktu sehingga data dapat digunakan secara lebih akurat pada tahap analisis selanjutnya. Contoh hasil standarisasi waktu ditunjukkan pada Tabel 4.

Tabel 4. Contoh Standarisasi Format Waktu pada Atribut *created_at*

Sebelum Standarisasi	Sesudah Standarisasi
Wed Jan 08 14:32:10 +0000 2026	2026-01-08 21:32:10
Sun Feb 15 23:47:35 +0000 2026	2026-02-16 06:47:35

Secara keseluruhan, hasil transform menunjukkan bahwa *pipeline* tidak hanya mengurangi data yang tidak memenuhi kriteria, tetapi juga menghasilkan data yang lebih terstruktur. Proses ini berjalan secara otomatis dan tetap stabil selama periode pengujian.

3. Load (Penyimpanan Data)

Seluruh data hasil ekstraksi dan transform berhasil dimuat ke dalam basis data. Sebanyak 8.662 entri tersimpan pada tabel *raw_data* sebagai data mentah, sedangkan 6.377 entri hasil pembersihan tersimpan pada tabel *clean_data*. Selama periode pengujian tujuh hari, proses penyimpanan berjalan tanpa kegagalan dan tidak ditemukan konflik pada atribut *id_str*. Hasil pengecekan menunjukkan tidak terdapat entri duplikat pada tabel *clean_data*, sehingga mekanisme pencegahan duplikasi berjalan sesuai rancangan. Selain itu, setiap siklus eksekusi harian berhasil menambahkan data baru tanpa menghapus data yang telah tersimpan sebelumnya. Hal ini menunjukkan bahwa skema penyimpanan berjalan konsisten selama periode pengujian dan mendukung keberlanjutan *pipeline* pada eksekusi berikutnya. Struktur tabel yang digunakan dalam penyimpanan data bersih ditunjukkan pada Tabel 5.

Tabel 5. Struktur Tabel *clean_data* pada Database

Nama Atribut	Tipe Data	Keterangan
<i>id</i>	INT	Primary key, auto increment
<i>id_str</i>	VARCHAR	ID unik tweet (UNIQUE)
<i>full_text_clean</i>	TEXT	Konten teks tweet hasil cleaning
<i>created_at</i>	VARCHAR	Waktu unggah (format WIB, disimpan sebagai teks)
<i>lang</i>	VARCHAR	Bahasa tweet
<i>source_file</i>	VARCHAR	Sumber file hasil <i>scraping</i>

Struktur tabel tersebut menunjukkan bahwa proses load tidak hanya berfungsi sebagai tahap penyimpanan, tetapi juga sebagai mekanisme pengendalian integritas data. Penambahan atribut *id* sebagai *primary key auto increment* mendukung manajemen baris data secara sistematis, sedangkan atribut *id_str* yang bersifat unik mencegah terjadinya duplikasi tweet.

Evaluasi Sistem

Evaluasi sistem dilakukan untuk menilai kualitas data hasil *scraping* serta kinerja teknis *pipeline* ETL selama eksekusi otomatis. Pengujian difokuskan pada tingkat relevansi data dan efisiensi penggunaan sumber daya sistem.

1. Hasil Uji Akurasi Hasil Scraping

Uji akurasi dilakukan untuk mengukur tingkat relevansi data hasil *scraping* berbasis kata kunci terhadap isu Program Makan Bergizi Gratis (MBG). Dari total 6.377 data pada tabel *clean_data* selama periode pengujian tujuh hari, dipilih 200 sampel untuk evaluasi. Relevansi ditentukan secara manual berdasarkan kesesuaian konteks isi tweet terhadap kebijakan MBG. Hasil validasi sampel ditunjukkan pada Tabel 6.

Tabel 6. Hasil Uji Akurasi (Precision) Berdasarkan Validasi Sampel

Kategori	Jumlah	Persentase
Relevan	191	95,5%
Tidak Relevan	9	4,5%
Total	200	100%

Berdasarkan hasil tersebut, nilai *precision* dihitung menggunakan Persamaan (1):

$$Precision = \frac{191}{200} \times 100\% = 95,5\%$$

Nilai *precision* sebesar 95,5% menunjukkan bahwa sebagian besar data yang dihasilkan *pipeline* telah sesuai dengan topik penelitian. Namun demikian, masih terdapat 4,5% data yang tidak relevan. Data noise tersebut umumnya muncul akibat penggunaan kata kunci dalam konteks yang ambigu atau tidak secara langsung membahas kebijakan MBG. Data yang tidak relevan menunjukkan bahwa noise pada hasil *scraping* didominasi oleh penggunaan kata kunci “MBG” dalam konteks yang tidak berkaitan dengan Program Makan Bergizi Gratis. Berdasarkan observasi terhadap sampel data, terdapat beberapa kategori utama noise.

Pertama, ambiguitas singkatan, di mana istilah “MBG” digunakan sebagai makna lain di luar konteks kebijakan, seperti pada tweet “inikah mbg my baby girl”, “udeh muak untung ada mbg my bini gue”, dan “mbg makan bulking gym kan maksudnyaa”. Kedua, konten informal atau candaan yang tidak mengandung informasi substantif, seperti “kuatin mbg saja wkwwkwkw”, “mbg agak halu dikit” dan “aku suka jokes mbg ini”. Dominasi noise jenis ini menunjukkan bahwa pendekatan berbasis keyword, khususnya penggunaan singkatan seperti “MBG”, rentan terhadap ambiguitas semantik. Hal ini menunjukkan bahwa semakin luas cakupan data yang diperoleh, semakin besar pula potensi munculnya data yang tidak relevan.

2. Hasil Uji Performa Sistem

Pengujian performa sistem dilakukan untuk mengevaluasi efisiensi *pipeline* ETL selama proses eksekusi. Hasil pengujian menunjukkan penggunaan sumber daya komputasi yang relatif rendah selama *pipeline* dijalankan. Ringkasan hasil pengujian performa sistem ditunjukkan pada Tabel berikut. Hasil pengujian performa sistem selama tujuh hari ditunjukkan pada Tabel 7.

Tabel 7. Hasil Pengujian Performa *Pipeline* ETL Selama Tujuh Hari

Hari Ke-	CPU (%)	RAM (MB)	Durasi Eksekusi	Status
1	0,30	145,2	34 menit 42 detik	Completed
2	0,29	145,7	32 menit 21 detik	Completed
3	0,36	146,8	34 menit 40 detik	Completed
4	0,36	147,2	56 menit 20 detik	Completed
5	0,76	146,7	36 menit 58 detik	Completed
6	0,76	146,4	33 menit 11 detik	Completed
7	0,64	146,4	29 menit 28 detik	Completed

Berdasarkan Tabel 7, penggunaan CPU dan RAM selama eksekusi tergolong rendah dan stabil, dengan rata-rata CPU sebesar 0,50% dan RAM sebesar 146,34 MB. Hal ini menunjukkan bahwa *pipeline* dapat berjalan tanpa membebani sistem. Durasi eksekusi berada pada kisaran 29 hingga 56 menit, yang dipengaruhi tidak hanya oleh jumlah data, tetapi juga oleh faktor eksternal seperti kondisi jaringan, respons server, dan proses pemuatan halaman saat scraping. Hal ini menunjukkan bahwa tahap *extract* menjadi komponen yang paling berpengaruh terhadap waktu eksekusi. Dibandingkan dengan pendekatan manual atau semi-manual, *pipeline* berbasis workflow orchestration menunjukkan keunggulan dalam hal konsistensi dan efisiensi proses. Penggunaan Prefect sebagai orchestration tool memungkinkan proses ETL berjalan secara otomatis, terjadwal, serta didukung oleh fitur monitoring dan mekanisme retry, sehingga *pipeline* tetap stabil meskipun terjadi gangguan sementara.

Meskipun scraping menjadi solusi alternatif atas keterbatasan akses API, pendekatan ini memiliki keterbatasan karena bergantung pada struktur halaman web yang dapat berubah sewaktu-waktu, serta keterbatasan dalam akses metadata dibandingkan API resmi. Oleh karena itu, scraping dalam penelitian ini diposisikan sebagai solusi praktis, bukan pengganti ideal dari API. Data hasil *pipeline* disimpan dalam basis data relasional pada tabel *clean_data*, sehingga dapat diakses kembali untuk analisis lanjutan, seperti analisis sentimen atau kebijakan. Selain itu, *pipeline* dapat dijalankan ulang dengan parameter kata kunci dan rentang waktu tertentu, sehingga memungkinkan pengambilan data secara berkelanjutan tanpa proses manual, serta mendukung replikasi oleh peneliti lain.

KESIMPULAN

Penelitian ini menghasilkan arsitektur *data pipeline* berbasis ETL yang terotomatisasi dan terorkestrasi menggunakan Prefect untuk pengumpulan data media sosial melalui *scraping*, serta menunjukkan bahwa integrasi proses *extract*, *transform*, dan *load* dalam workflow terjadwal mampu berjalan konsisten, efisien, dan menghasilkan data dengan tingkat relevansi tinggi. Secara konseptual, penelitian ini berkontribusi pada integrasi *scraping* data tidak terstruktur dengan *workflow orchestration* dalam satu *pipeline* terpadu, sekaligus menyediakan data yang tersimpan secara terstruktur dan dapat digunakan kembali untuk analisis lanjutan seperti analisis sentimen maupun analisis kebijakan. Selain itu, *pipeline* yang dirancang dapat digunakan kembali dan dijalankan ulang dengan parameter tertentu, sehingga mendukung pengumpulan data secara berkelanjutan dan replikasi penelitian. Namun, temuan ini terbatas pada konteks platform X dan studi kasus Program Makan Bergizi Gratis dalam periode pengujian tertentu, sehingga tidak dapat digeneralisasi secara luas; oleh karena itu, penelitian selanjutnya disarankan menguji arsitektur ini pada berbagai platform, skenario, dan skala yang lebih besar untuk meningkatkan validitas dan keberlanjutan sistem.

DAFTAR PUSTAKA

- Amanda, A. F., & Wasposito, B. (2025). Implementation of Extract, Transform, Load (ETL) Method for Sales Data Visualization Using Google Looker Studio. *Jurnal Ilmiah Media Sisfo*, 19(2), 124–134. <https://doi.org/10.33998/mediasisfo.2025.19.2.2547>
- Amirullah, F., Alam, S., Sulisty, M. I. (2023). Analisis Sentimen Terhadap Kinerja KPU Menjelang Pemilu 2024 Berdasarkan Opini Twitter Menggunakan Naive Bayes. *Jurnal Ilmiah Teknik dan Ilmu Komputer*, 2(3), 69–76.
- Andriansyah, D. (2022). Implementasi Extract-Transform-Load (ETL) Data Warehouse Laporan Harian Pool. *Jurnal Teknik Informatika*, 8(2), 45–49. <https://doi.org/10.51998/jti.v8i2.486>
- Baiochi, J., Bernardi, F., Yamada, D., Felix, T., Ferraz, V., Lima, V., & Alves, D. (2025). Development of an Infrastructure and Computational Pipeline for Analyzing Rare Disease Research Data: A Case Study in the State of Sao Paulo, Brazil. *Procedia Computer Science*, 256, 1341–1349. <https://doi.org/10.1016/j.procs.2025.02.247>
- Barutu, S., Silalahi, R. J., Tambunan, Y., Lipipian, P. A., & Yohannes, Y. (2025). Analisis Sentimen Masyarakat Mengenai Revisi UU TNI di X Menggunakan Metode Naive Bayes. *Jurnal Informatics and Computer Engineering Journal*, 5(2), 105–114. <https://doi.org/10.31294/icej.v5i2.9544>
- Bintang, A., Hananto, A. L., & Hananto, A. (2025). Telegram BOT Application Development Integration with Google Sheets for Sending Service Reporting. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, 4(3), 2208–2214. <https://doi.org/10.59934/jaiea.v4i3.1136>
- Blakey, E. (2024). The Day Data Transparency Died: How Twitter/X Cut Off Access for Social Research. *Contexts*, 23(2), 30–35. <https://doi.org/10.1177/15365042241252125>
- Brewer, R., Westlake, B., Arauza, O., & Hart, T. (2021). The Ethics of Web Crawling and Web Scraping in Cybercrime Research: Navigating Issues of Consent, Privacy, and Other Potential Harms Associated with Automated Data Collection. *Researching Cybercrimes: Methodologies, Ethics, and Critical Approaches*, 435–456. https://doi.org/10.1007/978-3-030-74837-1_22
- Chanda, D. (2024). Automated ETL Pipelines for Modern Data Warehousing: Architectures, Challenges, and Emerging Solutions. *The Eastasouth Journal of Information System and Computer Science*, 1(3), 209–212. <https://doi.org/10.58812/esiscs.v1i03>
- Fadhlullah, N., Setiawansyah, S., & Surahman, A. (2022). Penerapan Teknologi Web Scraping Sebagai Pengumpulan Data Covid-19 Di Provinsi Lampung. *Jurnal Informatika Dan Rekayasa Perangkat Lunak (JATIKA)*, 3(1), 25–30. <https://doi.org/10.33365/jatika.v3i1.1841>
- Firdonsyah, A., & Tyas, Z. A. (2021). Peningkatan pengetahuan remaja tentang bijak menggunakan media sosial di SMU Muhammadiyah Gombong. *Jurnal Pengabdian Masyarakat*, 1(1), 38–45. <https://doi.org/10.31101/hayina.2227>
- Flores, V. A., Permatasari, P. A., & Jasa, L. (2020). Penerapan Web Scraping Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan Keyword. *Majalah Ilmiah Teknologi Elektro*, 19(2), 157. <https://doi.org/10.24843/mite.2020.v19i02.p06>
- Ghani, R. A., & Kurniawan, R. (2024). Implementasi Extract, Transform, Load Process Pada Perancangan Data Warehouse Terkait Kualitas Pendidikan Di Kabupaten Serang. In *Jurnal Mahasiswa Teknik Informatika*, 8(2), 2083–2090. <https://doi.org/10.36040/jati.v8i2.9081>
- Javier, N., Satoto, B. D., & Negara, Y. D. P. (2025). Implementasi Teknik Web Scraping Untuk Pengumpulan Data Laporan Keuangan Perusahaan Di Bursa Efek Indonesia (IDX). In *Jurnal Mahasiswa Teknik Informatika*, 9(2), 2789–2795.

- Kiftiyah, A., Palestina, F. A., Abshar, F. U., & Rofiah, K. (2025). Program Makan Bergizi Gratis (MBG) dalam Perspektif Keadilan Sosial dan Dinamika Sosial – Politik. *Pancasila: Jurnal Keindonesiaan*, 5(1), 101–112. <https://doi.org/10.52738/pjk.v5i1.726>
- Kridono, P. K., & Maukar. (2025). Pengembangan Dashboard Analitik *Real-time* Berbasis BPMN Camunda untuk Optimalisasi Proses Produksi dan Distribusi di Industri Maklon. *Decode: Jurnal Pendidikan Teknologi Informasi*, 5(2), 429–442. <https://doi.org/10.51454/decode.v5i2.1126>
- Kupfer, A. (2024). Nonrandom Tweet Mortality and Data Access Restrictions: Compromising the Replication of Sensitive Twitter Studies. *Political Analysis*, 32(4), 493–506. <https://doi.org/10.1017/pan.2024.7>
- Lediwara, N., Bimorogo, S. D., Heikmakhtiar, A. K., Gainal, I. J., Karomah, S. N., & Fahmi, K. (2024). Sentiment Analysis About UNHAN RI Through Twitter Social Media. *Antivirus : Jurnal Ilmiah Teknik Informatika*, 18(2), 212–221. <https://doi.org/10.35457/antivirus.v18i2.3902>
- Maskhuri, A., & Hardiani, T. (2025). Analisis Sentimen Pengguna pada Aplikasi Tokopedia Menggunakan Algoritma Convolutional Neural Network. *Technology and Science (BITS)*, 6(4). <https://doi.org/10.47065/bits.v6i4.6923>
- Mnsen, H. K. I., Purnomosidi, B., Kartadie, R., & Kurnaedi, D. (2024). Data Pipeline Architecture For Academic Information System At Akademi Teknik Biak. *Journal of Intelligent Software Systems*, 3(1), 1-6. <https://doi.org/10.26798/jiss.v3i1.1335>
- Murtefeldt, R., Paik, S., Alterman, N., Kahveci, I., & West, J. D. (2024). RIP Twitter API: A Eulogy to Its Vast Research Contributions. *Cornell University*, 1, 1-34.
- Octiva, C. S., Fajri, T. I., Sulistiarini, E. B., Suharjo, S., & Nuryanto, U. W. (2024). Penggunaan Teknik Data Mining untuk Analisis Perilaku Pengguna pada Media Sosial. *Jurnal Minfo Polgan*, 13(1), 1074–1078. <https://doi.org/10.33395/jmp.v13i1.13936>
- 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>
- Trezza, D. (2023). To Scrape or not to Scrape, This is Dilemma. The post-API Scenario and Implications on Digital Research. *Frontiers in Sociology* 8, 1-8. <https://doi.org/10.3389/fsoc.2023.1145038>
- Artika, R. D., Rafifah, N. N., Dina, P. A., & Rozikin, C. (2025). Perancangan Data Pipeline Untuk Analisis Pola Perjalanan Dan Permintaan Layanan Transjakarta. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(3S1). <https://doi.org/10.23960/jitet.v13i3S1.8051>
- Staneva, V., Butala, S., Setiawan, L. (Don), & Lee, W.-J. (2024). Echodataflow: Recipe-based Fisheries Acoustics Workflow orchestration. *Proceedings of the 23rd Python in Science Conference*, 349–365. <https://doi.org/10.25080/jxdk4427>
- Suhendra, S., & Pratiwi, F. S. (2024). Peran Komunikasi Digital dalam Pembentukan Opini Publik: Studi Kasus Media Sosial. *Iapa Proceedings Conference*, 293. <https://doi.org/10.30589/proceedings.2024.1059>
- Widcaksono, A., & Silmina, E. P. (2023). Rancang Bangun Sistem Informasi Back Office Aplikasi Telemedicine Sebagai Media Pembelajaran. *Jurnal Informatika Terpadu*, 9(2), 119–125. <https://doi.org/10.54914/jit.v9i2.973>
- Yasmin, J., Wang, J., Tian, Y., & Adams, B. (2024). An empirical study of developers' challenges in implementing Workflows as Code: A case study on Apache Airflow. *Journal of Systems and Software*, 219, 112248. <https://doi.org/10.1016/j.jss.2024.112248>

- Yuniar, E., Utsalinah, D. S., & Wahyuningsih, D. (2022). Implementasi Scrapping Data Untuk Sentiment Analysis Pengguna Dompot Digital dengan Menggunakan Algoritma Machine Learning. *Jurnal Janitra Informatika Dan Sistem Informasi*, 2(1), 35–42. <https://doi.org/10.25008/janitra.v2i1.145>
- Zulaika, N., Lestari, D., & Istiqomah, H. (2025). Tantangan Implementasi dan Akuntabilitas Anggaran Program Makanan Bergizi Gratis (MBG) Berdasarkan Instruksi Presiden Nomor 1 Tahun 2025. *Jurnal Penelitian Ilmiah Multidisipliner*, 1(03), 426–435.