JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

Pipeline ETL Big Data sebagai Solusi Integrasi Data Perguruan Tinggi dengan Evaluasi Validitas Otomatis

Iis Pradesan[1], Dorie Pandora Kesuma[2]
[1,2] Sistem Informasi, Universitas Multi Data Palembang
Email: [1]iis@mdp.ac.id, [2]dpkesuma@mdp.ac.id
Email Penulis Korespondensi: iis@mdp.ac.id

Abstrak
Data akademik di perguruan tinggi, seperti mahasiswa, dosen, mata kuliah, kelas, dan nilai, tersebar di berbagai sumber heterogen, sehingga menimbulkan tantangan besar pada integrasi, dan standar validitas data. Kondisi ini berpotensi menghambat pelaporan, akreditasi, maupun pengambilan keputusan strategis berbasis data. Penelitian ini bertujuan merancang dan mengimplementasikan pipeline *Extract, Transform, Load* (ETL) berbasis *big data* untuk mengatasi permasalahan tersebut. Metode penelitian meliputi studi pendahuluan, pengumpulan data, pemetaan rule validasi, dan implementasi *pipeline* menggunakan *Apache NiFi* serta *Hadoop Distributed File System* (HDFS). Hasil implementasi menunjukkan *pipeline* mampu mengekstrak data multi-sumber secara otomatis, menerapkan validasi berbasis aturan (*length, nullable, reference, min-max*), dan memuat data tervalidasi ke *cluster* HDFS dengan *high availability* dan *fault tolerance*. Uji coba menemukan sekitar 8% data *error* berhasil diisolasi, sementara tingkat validitas data pada entitas utama mencapai lebih dari 90%. Kontribusi penelitian ini terletak pada integrasi multi sumber data akademik, penerapan standar, mekanisme otomatis penanganan data invalid, serta pemanfaatan HDFS sebagai penyimpanan terdistribusi. *Pipeline* yang dihasilkan dapat menjadi *blueprint* praktis untuk perguruan tinggi di Indonesia dalam mengukur kualitas data dan mendukung tata kelola berbasis big data.
Kata Kunci: Pipeline, ETL, Big Data, Data Terstruktur, Validasi Data, Perguruan Tinggi

Abstract
Academic data in higher education institutions encompassing students, lecturers, courses, classes, and grades are often dispersed across various heterogeneous sources. This fragmentation poses significant challenges to integration and data validity standards. Such conditions potentially impede reporting, accreditation, and data-driven strategic decision-making. This study aims to design and implement a Big Data-based Extract, Transform, Load (ETL) pipeline to address these issues. The research methodology includes a preliminary study, data collection, validation rule mapping, and pipeline implementation utilizing Apache NiFi and the Hadoop Distributed File System (HDFS). The implementation results demonstrate the pipeline's capability to automatically extract multi-source data, apply rule-based validation (length, nullable, reference, and min-max), and load validated data into an HDFS cluster featuring high availability and fault tolerance. Testing revealed that approximately 8% of erroneous data was successfully isolated, while the data validity rate for primary entities exceeded 90%. The contributions of this research lie in the integration of multi-source academic data, the application of standards, the automated mechanism for handling invalid data, and the utilization of HDFS as distributed storage. The resulting pipeline serves as a practical blueprint for higher education institutions in Indonesia to assess data quality and support Big Data-based governance.
Keywords*: Pipeline, ETL, Big Data, Data Structured, Data Validity, Higher Education Institution (at least 5 words related to the research content separated by commas)*

1. PENDAHULUAN

Perguruan tinggi saat ini menghadapi tantangan besar dalam mengelola data yang semakin kompleks, heterogen, dan berukuran besar. Data tersebut mencakup seluruh kegiatan tridarma, mulai dari pengajaran, penelitian, pengabdian kepada masyarakat, hingga aspek pendukung seperti keuangan, perpustakaan dan pemasaran. Data berasal dari berbagai sumber, baik sistem basis data operasional dan dokumen *spreadsheet* yang tersebar di berbagai unit kerja. Kondisi ini menyebabkan fragmentasi, redudansi, serta inkonsistensi data, sehingga sulit dimanfaatkan untuk pelaporan maupun pengambilan keputusan strategis. Konsekuensinya, banyak perguruan tinggi menghadapi masalah serius seperti keterlambatan pelaporan ke Pangkalan Data Pendidikan Tinggi (PDDIKTI), kesalahan data dalam proses akreditasi, hingga kurang optimalnya kebijakan akademik karena minimnya data berkualitas.

Dalam konteks *big data*, integrasi data menjadi fondasi penting untuk mewujudkan *data driven decision making* di lingkungan pendidikan tinggi. Tantangan utama tidak hanya terkait dengan jumlah/*volume* data, tetapi juga variasi/*variety* format dan kecepatan/*velocity* pertumbuhan data yang harus diolah secara efisien. Teknologi *Extract, Transform, Load* (ETL) berbasis big data menawarkan solusi yang relevan, karena mampu melakukan akuisisi, transformasi, dan pemuatan data dalam jumlah besar secara otomatis, terstruktur, dan sesuai aturan validasi yang ditetapkan.

Berbagai penelitian sebelumnya menunjukkan bahwa penerapan big data di bidang pendidikan mampu memberikan kontribusi signifikan dalam mendukung pengambilan keputusan akademik dan manajerial [1]. Penggunaan teknologi data

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

warehouse berbasis *big data* juga terbukti meningkatkan efisiensi pengolahan data perguruan tinggi [2]. Selain itu, rancangan arsitektur *big data ingestion* khusus untuk lingkungan pendidikan telah diajukan sebagai solusi integrasi data multi sumber [3]. Dalam konteks proses ETL, penelitian terkini mengidentifikasi tantangan utama pada tahapan ekstraksi, transformasi, dan pemuatan data dalam skala besar, serta menekankan pentingnya optimalisasi *pipeline* ETL untuk *text mining* maupun analitik lanjutan [4].

Kajian lain menekankan aspek kualitas *pipeline* data, faktor kegagalan proses ETL [5], bahkan dalam tinjauan sistematis, perkembangan penelitian big data selama 15 tahun terakhir menegaskan bahwa *pipeline* ETL merupakan komponen fundamental yang tidak terpisahkan dari ekosistem *big data* [6]. Tren terbaru menunjukan bahwa *pipeline* berbasis *serverless* telah ditawarkan sebagai tren masa depan dalam memproses data skala besar dengan fleksibilitas yang lebih tinggi [7].

Khusus pada lingkungan perguruan tinggi, *pipeline* ETL sudah pernah diimplementasikan mendukung analitik penerimaan mahasiswa baru [8]. Hasil penelitian tersebut menunjukkan bahwa *pipeline* ETL tidak hanya berfungsi untuk integrasi, tetapi juga bermanfaat dalam analisis prediktif. Hal ini diperkuat dengan studi terkait *educational big data* yang menegaskan bahwa keberhasilan pemanfaatan *big data* di bidang pendidikan terletak pada kemampuan mengekstraksi makna dari data terstruktur [9].

Tabel 1. Penelitian sejenis

Penelitian	Hasil	GAP Penelitian
Data Pipeline Architecture for Academic Information System Akademi Teknik Biak. [10]	Implementasi *pipeline* ETL untuk data akademik dengan format *csv* dan berjalan di atas *apache spark*.	Sumber data sebatas *spreadsheet*, dan belum mengimplementasikan *Big Data distributed system.*
Automated ETL *Pipelines for Modern Data Warehousing*: *Architectures, Challenges, and Emerging Solutions*. [11]	Merancang arsitektur *Cloud-Native* untuk proses ETL terotomatisasi, termasuk peran *metadata driven* automation dan AI dalam data *warehousing*	Hanya sebatas rancangan arsitektur belum sampai pada implementasi dan berbasis aturan.
Pemanfaatn Big Data dalam Sistem Informasi untuk Pengambilan Keputusan Strategis. [12]	Eksplorasi integrasi Big Data ke dalam sistem informasi untuk sebagai pendukung keputusan strategis	Hanya sebatas eksplorasi teori *Big Data* pada SI organisasi, belum sampai pada implementasi.
ETL *Proecess Automation*: *Tools* and *Techniques*. [13]	Otomatisasi ETL umum	Sebatas ETL dan tidak ada integrasi dengan *Big Data*.
Managing Data Quality and Consistency in Real-Time ETL for Streaming Applications: A Comparative Analysis of Modern ETL Frameworks. [14]	Kerangka evaluasi ETL *streaming* berbasis kualitas data yang terintegrasi dengan ekosistem *big data*	Kurangnya pembahasan integrasi aturan kualitas data dan implementasi nyata pada organisasi.
Maximizing ETL efficiency: Patterns for high-volume data. [15]	*Efisiensi ETL serverless*.	Tidak ada integrasi standar kualitas data.

Dari tinjauan tersebut dapat diidentifikasikan *gap* penelitian yaitu perlunya penelitian yang mengintegrasikan multi sumber dengan aturan validasi serta penyimpanan terdistribusi menggunakan *big data*. Oleh karena itu, penelitian ini bertujuan mengimplementasikan *pipeline* ETL *big data* yang mampu menghasilkan *single source of truth* data akademik, sekaligus menjadi fondasi untuk analitik lanjutan dan pelaporan institusional.

Penelitian ini menawarkan beberapa kontribusi utama yang membedakannya dari studi sebelumnya: 1. Integrasi Multi Sumber Data: *Pipeline* mampu menggabungkan data dari *spreadsheet* (.xls/.csv) dan basis data relasional (MySQL) dalam satu alur otomatis; 2. *Rule Validasi* : Aturan validasi (*length*, *nullable*, *reference*, *min-max*); 3. Automasi Penanganan Data Invalid: *Pipeline* tidak hanya memfilter, tetapi juga mengisolasi data tidak valid ke direktori khusus sebagai bahan perbaikan kemudian; 4. Penyimpanan Terdistribusi HDFS: Pemanfaatan *cluster node* dengan *replication* mendukung *high availability* dan *fault tolerance*; 5. *Blueprint* Praktis: Model *pipeline* dapat diadopsi langsung oleh perguruan tinggi di Indonesia untuk persiapan akreditasi, audit data, dan analitik strategis berbasis *big data*.

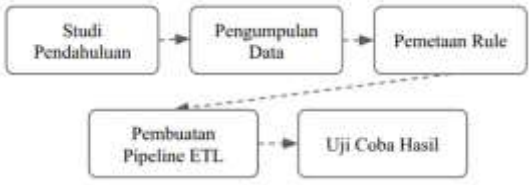
2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan dengan beberapa tahapan seperti pada Gambar 1 di bawah ini yang terdiri dari :

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

a. Berdasarkan literatur dari penelitian serupa dan didukung dengan referensi toeritis lainnya terkait teknologi *big data*, *pipeline*, proses ETL serta wawancara dengan perguruan tinggi, ditemukan masalah yang sering terjadi antara lain keterpisahan data, lemahnya validitas, dan kebutuhan integrasi. Tujuan penelitian ini adalah merancang *pipeline* ETL berbasis *big data* yang dapat mengintegrasikan data terstruktur multi sumber dan tervalidasi otomatis;
b. Data dikumpulkan dari dua perguruan tinggi melalui wawancara dan observasi. Lingkup penelitian dibatasi pada lima entitas utama: Mahasiswa, Dosen, Kelas, Mata Kuliah, dan Nilai. Total jumlah *field* tiap entitas ditampilkan pada Tabel 2. Variasi jumlah *field* ini menambah kompleksitas *pipeline* karena tiap entitas memiliki aturan validasi berbeda;
c. Pada dasarnya *rule* data akan menyesuaikan kebutuhan perguruan tinggi, namun dalam penelitian ini diberikan beberapa *rule* yang relevansinya pada validitas data akademik sesuai aturan teknis dari sistem Pangkalan Data Pendidikan Tinggi (PDDIKTI), dengan harapan dapat membatu mengurangi kesalahan saat perguruan tinggi melakukan sinkronisasi data. Adapun *rule* dibedakan menjadi 4 kategori yaitu *length*, *minimum value*, *maximum value*, *nullable*, dan *referensi*. *Rule Length* digunakan untuk membatasi jumlah karakter, *minimum value* digunakan untuk batas bawah dari nilai angka, *maximum value* digunakan untuk batas atas dari nilai angka, *nullable* digunakan ntuk memastikan nilai harus diisi atau boleh dikosongkan dan referensi digunakan untuk memastikan nilai sesuai kategori yang telah ditentukan;
d. Ini merupakan tahapan inti dari penelitian ini, di dalamnya terdapat *proses* ETL yang dibagun di atas ekosistem *Hadoop* yaitu *Apache NiFi*, dan *Apache Hadoop*. *Pipeline* juga akan didukung antarmuka sehingga dapat berinteraksi dengan pengguna, yang pada akhirnya *pipeline* akan manghasilkan report berupa *dashboard* berbasi web guna memvisualisasikan persentase data valid atau invalid berikut dengan detailnya.



Gambar 1. Tahapan Penelitian

3. HASIL DAN PEMBAHASAN

3.1 Arsitektur Pipeline ETL

Arsitektur *pipeline* ditunjukkan pada Gambar 2. Proses ETL dimulai dari ekstraksi data (*spreadsheet* dan *database*), validasi berbasis *rule*, hingga pemuatan ke HDFS. *Processor* utama dipetakan langsung ke fungsi ETL yaitu *GetFile* / *QueryDatabaseTable* (*Extract*), *RouteOnAttribute* / *ReplaceText* (*Transform*), dan *PutHDFS* (*Load*).

Gambar 2. Arsitektur *Pipeline* ETL

Data kemudian dimuat/*load* ke dalam HDFS untuk kemudian didistribusikan pada tiga data *node* dalam satu *cluster big data* serta menggunakan tiga replikasi untuk tiap datanya, guna memastikan ketersediaan data dan *fault telorance* yang merupakan salah satu konsep penyimpanan pada *big data*.

3.2 Tahapan Proses ETL

3.2.1 Extract

Proses dimulai dari *extract* data yang berfungsi mengambil data dari dua pilihan sumber data terstruktur yang disiapkan, yaitu *spreadsheet* dan *relational database* yang terhubung secara *realtime* dengan *database* MySQL. Selain sumber data akademik, ekstrak data juga dilakukan untuk *spreadsheet rule*. Total *field* per entitas ditunjukkan pada Tabel 2.

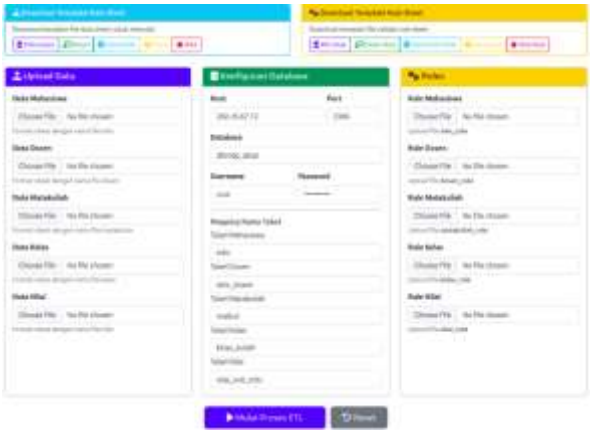
Tabel 2. Data Akademik

Data	Jumlah Field Mandatory

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

Mahasiswa	27
Dosen	18
Mata Kuliah	8
Kelas	11
Nilai	10

Antarmuka pengguna yang digunakan untuk menampilkan sumber data serta aturan validasi dapat dilihat pada Gambar 3. Pada bagian sumber data pengguna diminta mengunggah sumber data Mahasiswa, Dosen, Mata Kuliah, Kelas, dan Nilai, atau menggunakan konfigurasi koneksi ke MySQL. Sementara pada bagian *rule*, pengguna dapat melihat dan menetapkan aturan validasi sesuai standar yang ditetapkan, seperti panjang karakter, nilai minimum–maksimum, *nullable*, dan referensi.



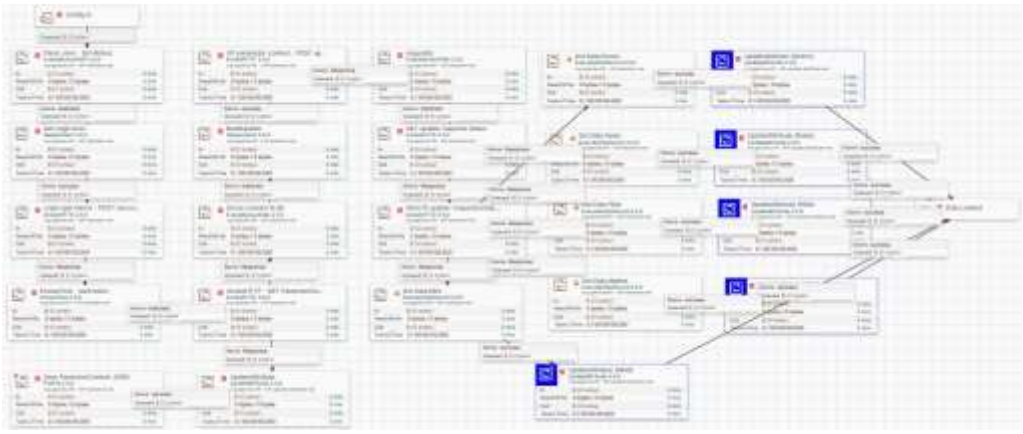
Gambar 3. Antarmuka Pengguna Sumber Data dan *Rule*

Pada Gambar 4 menunjukkan *group processor* pada tahap *Extract* di *Apache NiFi*, *processor* seperti *GetFile* dan *QueryDatabaseTable* berfungsi mengimpor data mentah secara otomatis ke dalam *pipeline*. *Group processor* ini memastikan seluruh entitas utama dapat diekstrak secara konsisten dan *realtime* sebagai langkah awal sebelum dilakukan validasi.

Gambar 4. *Extract Pipeline*

Alur *processor* pada Gambar 5 memperlihatkan tahap *Extract* yang secara khusus digunakan untuk mengambil data dari basis data relasional. Tahapan diawali dengan konfigurasi parameter koneksi *database*, termasuk nama tabel, kolom, serta kredensial yang disiapkan melalui *processor Set Attribute* dan *ReplaceText*. Untuk memastikan keamanan dan validitas koneksi, *pipeline* juga memanfaatkan *InvokeHTTP* serta *EvaluateJsonPath* dalam proses autentikasi dan pengelolaan parameter. Setelah konfigurasi dan autentikasi selesai, *processor Get Data* dijalankan untuk mengekstraksi entitas utama dari *database*. Dengan rancangan ini, *pipeline* mampu melakukan pengambilan data langsung dari sumber *database* dengan cara yang otomatis, aman, dan konsisten.

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi



Gambar 5. *Extract* DB *Pipeline*

Data yang berhasil diambil dari tiap entitas kemudian diberi label melalui *processor UpdateAttribute*. Dengan struktur modular tersebut, *pipeline* dapat dioperasikan ulang tanpa konfigurasi manual berulang, serta mudah diperluas jika di kemudian hari diperlukan penambahan tabel atau entitas baru.

3.2.2 Transform

Proses berikutnya yaitu *transform* digunakan untuk memproses data dengan menggunakan *processor GetFile*, *RouteOnAttribute*, *PutFile* dan lain-lain untuk kemudian di komparasi dengan *rule* sebagai pemerikaan validitas data yang telah ditentukan seperti pemeriksaan panjang *field*, nilai minimum, nilai maximum, *nullability*, dan nilai referensi untuk setiap kolom. *Transform* berbasis aturan ini bertujuan untuk memastikan konsistensi dan kualitas data.

a. *Data Rule Stage*

Rule data yang dipetakan pada tahap ini merupakan representasi dari kebutuhan validitas data akademik sesuai dengan standar teknis Pangkalan Data Pendidikan Tinggi (PDDIKTI). Penerapan aturan ini pada *pipeline* ETL berfungsi sebagai gerbang kualitas data yang otomatis, memastikan setiap entitas data memenuhi standar minimum sebelum disimpan ke dalam HDFS. Mekanisme ini secara fundamental mengurangi data *cleansing* manual dan meningkatkan kepercayaan terhadap data yang digunakan untuk pengambilan keputusan strategis. Aturan yang ditampilkan belum mencakup keseluruhan, masih banyak *field* lain dengan aturan validasi tambahan yang dapat digunakan atau disesuaikan sesuai kebutuhan masing-masing perguruan tinggi.

Aturan validasi data Mahasiswa pada Tabel 3 hanya menampilkan beberapa *field* contoh, seperti NIM, Nama, Agama, dan Penerima KPS. Pengecekan *length* digunakan untuk memastikan format data sesuai standar, aturan *nullable* menjaga kelengkapan informasi wajib, sedangkan aturan *reference* memastikan nilai merujuk pada kode terstandar PDDIKTI (misalnya ‘0’ atau ‘1’ untuk Penerima KPS, serta daftar nilai baku untuk Agama).

Tabel 3. *Rule* Data Mahasiswa

Kolom	Len	Min	Max	Null	Ref
Nama	11			T	
Agama				T	Islam; Kristen; Katolik; ..
Penerima KPS	1	0	1	T	0;1
...	...	...	...	...	...

Validasi data Dosen berorientasi pada kelayakan akademik staf pengajar. Fokus utama terletak pada atribut NIDN yang wajib diisi dan memiliki panjang karakter 10, menjadikannya identitas unik nasional. Selain itu, aturan *Reference* pada Jabatan Fungsional (Asisten Ahli, Lektor, dsb.) adalah kritikal. Setiap data Dosen yang masuk harus divalidasi terhadap daftar jabatan fungsional yang diakui untuk menjamin konsistensi data kepegawaian.

Penetapan batasan *Minimum* dan *Maximum* untuk atribut Masa Kerja (antara 0 hingga 99) memastikan bahwa data memiliki validitas logis dan dapat digunakan untuk menghitung Beban Kerja Dosen (BKD) atau proyeksi karir lainnya. Penerapan aturan ini secara otomatis dalam pipeline ETL Big Data memastikan bahwa data sumber daya manusia inti perguruan tinggi adalah akurat, yang mendukung sistem manajemen sumber daya yang efisien dan memitigasi risiko kesalahan dalam proses akreditasi institusi. *Rule* data dosen tersebut datap dilihat pada Tabel 4 berikut.

Tabel 4. Rule Data Dosen

Kolom	Len	Min	Max	Null	Ref

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

Nama Dosen				T	
NIDN	10			T	
Jabatan Fungsional				T	Asisten Ahli; Lektor; Guru Besar
Masa Kerja	2	0	99	T	
...	...	...	...	...	...

Rule validasi Mata Kuliah pada Tabel 5 dirancang untuk menjamin integritas kurikulum dan konsistensi katalog mata kuliah. Aturan *Length* dan *Nullable* diterapkan pada Kode dan Nama mata kuliah untuk memastikan setiap mata kuliah terdefinisi dengan jelas. Kepatuhan terhadap PDDIKTI diwujudkan melalui validasi atribut Jenis (misalnya, wajib, pilihan) dan SKS TM (Sistem Kredit Semester Tatap Muka). Konsistensi pada entitas Mata Kuliah ini sangat penting, karena merupakan dasar bagi seluruh transaksi akademik selanjutnya (Kelas dan Nilai). Jika data Mata Kuliah tidak valid, seluruh analisis hulu dan hilir akan terpengaruh. Oleh karena itu, pipeline ETL memastikan bahwa hanya mata kuliah yang terstruktur dengan SKS dan jenis yang benar yang dimuat ke HDFS, menyediakan fondasi data yang solid untuk analisis pemetaan kurikulum dan efektivitas pengajaran.

Tabel 5. Rule Data Mata Kuliah

Kolom	Len	Min	Max	Null	Ref
Kode	2			T	
Nama	200			Y	
Jenis	1			Y	A;B;C
SKS TM	2			Y	
...	...	...	...	...	...

Validasi data Kelas pada Tabel 6 bertujuan memastikan konsistensi operasional perkuliahan. Atribut utama yang divalidasi mencakup Nama Kelas, Semester, dan Mode Kuliah. Pemeriksaan *length* pada *field* Semester menjaga keseragaman format waktu (misalnya 5 karakter: 20251), sedangkan aturan *nullable* memastikan kelengkapan data penawaran kelas. Aturan *reference* pada Mode Kuliah (O: Online, F: Tatap Muka, M: Hybrid) penting untuk mencatat metode pembelajaran yang digunakan. Dengan validasi ini, data kelas menjadi lebih terstruktur dan dapat dimanfaatkan dalam lingkungan big data untuk menganalisis hubungan antara metode kuliah dengan kinerja akademik mahasiswa, sehingga mendukung strategi pendidikan berbasis bukti. Validasi ini juga membantu mengurangi potensi kesalahan input data dari unit akademik. Selain itu, penerapan standar seragam memungkinkan integrasi lintas sistem menjadi lebih mudah dan akurat.

Tabel 6. Rule Data Kelas

Kolom	Len	Min	Max	Null	Ref
Nama				T	
Semester	5			T	
Mode Kuliah	1			Y	O;F;M
...	...	...	...	...	...

Rule validasi untuk entitas data nilai pada Tabel 7, yang mencakup Nilai Angka harus memiliki panjang maksimal 5 karakter dengan rentang nilai antara 0 hingga 100 dan tidak boleh kosong, sedangkan Nilai Indeks dibatasi antara 0 hingga 4. Selain itu, Nilai Huruf harus sesuai dengan referensi nilai huruf seperti A, A- hingga E. Aturan-aturan ini dirancang untuk memastikan bahwa data nilai yang dimasukkan ke dalam sistem telah memenuhi standar kualitas dan konsistensi yang diperlukan sebelum dimuat ke dalam sistem penyimpanan terdistribusi.

Tabel 7. Rule Data Kelas

Kolom	Len	Min	Max	Null	Ref
Id Mahasiswa				T	
Kelas				T	
Nilai Angka	5	0	100	Y	
Nilai Indeks	5	0	4	Y	
Nilai Huruf	2			Y	A;A-;...;E
...	...	...	...	...	...

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

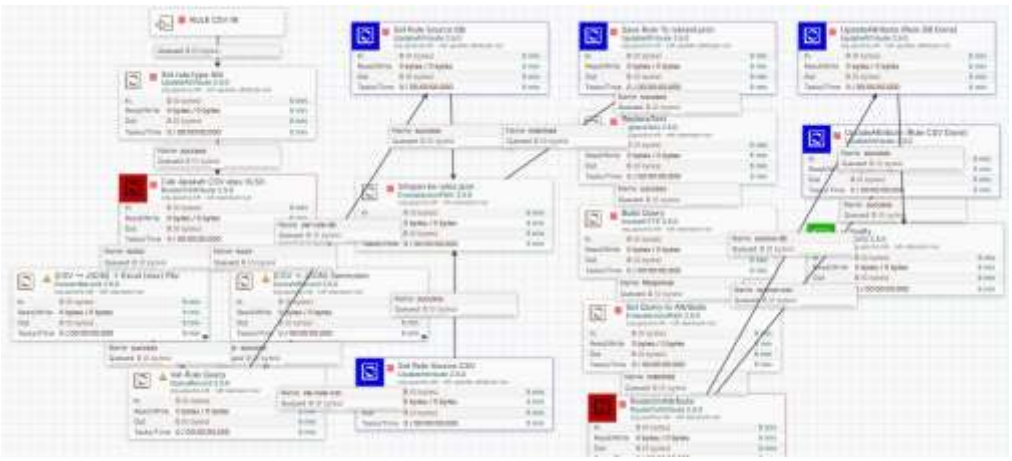
b. Transform Process Pipeline

Alur utama tahap transformasi data di *Apache NiFi* terdiri atas penentuan *rule*, identifikasi sumber data, serta proses validasi. *Processor UpdateAttribute* digunakan untuk menetapkan sumber data (CSV maupun DB), kemudian masing-masing aliran diarahkan ke *processor group* validasi sesuai asal datanya, yaitu "Proses Validasi Sumber Data Sheet (CSV)" dan "Proses Validasi Sumber Data DB". Sebelumnya, rule validasi ditentukan dalam *processor group* "Penentuan Rule/Referensi", yang kemudian menjadi acuan bagi kedua jalur validasi data. Dengan desain ini, pipeline memungkinkan rule ditetapkan secara terpusat dan diterapkan otomatis pada data hasil ekstraksi *spreadsheet* maupun *database*, sehingga konsistensi validasi antar-sumber tetap terjaga sebagaimana ditunjukkan pada Gambar 6.

Gambar 6. *Transform Stage*

Alur pada gambar 7 memperlihatkan tahap penerapan *rule* validasi terhadap data dari berbagai sumber. Data *CSV* yang dibaca melalui *GetFile* atau *ConvertExcelToCSV* terlebih dahulu diberi atribut sumber dengan *UpdateAttribute*, kemudian dikonversi ke format *JSON* menggunakan *ConvertRecord* agar seragam dengan format *rule*. Pada sisi lain, data dari *database* diambil melalui *ExecuteSQLRecord* dan juga dipersiapkan dengan *Set Rule Source DB*. Rule validasi sendiri diambil dari *file* referensi dengan *GetFile* atau hasil *query*, lalu diformat ke *JSON* sebelum digunakan.

Setelah data dan *rule* tersedia, *pipeline* menggunakan *ReplaceText*, *BuildQuery*, dan *SetQueryAttribute* untuk menyusun pernyataan validasi yang sesuai dengan tipe data. *Processor RouteOnAttribute* kemudian memisahkan data yang lolos validasi dan yang melanggar *rule*. Data valid diteruskan ke jalur utama untuk diproses lebih lanjut, sementara data tidak valid diarahkan ke folder khusus sebagai catatan perbaikan. Sebagai tambahan, *Notify* digunakan untuk memberi umpan balik status *rule* yang telah dijalankan. Alur modular ini memungkinkan integrasi *rule* validasi diterapkan secara otomatis baik pada data *CSV* maupun *database*.



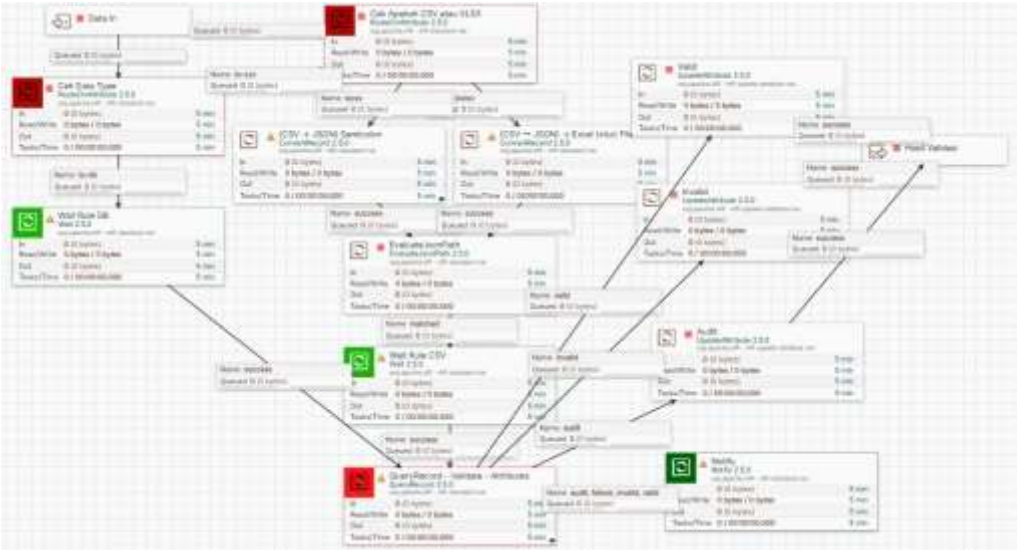
Gambar 7. *Transform Rule Stage*

Alur *pipeline* berikutnya menunjukkan bagaimana data dari *spreadsheet* divalidasi menggunakan *rule* yang sudah ditentukan. Proses dimulai dengan *Backup File Asli* agar salinan data mentah tetap tersimpan. Selanjutnya, *RouteOnAttribute* digunakan untuk mengecek format file apakah **.csv* atau **.xlsx*. *File* kemudian dikonversi ke format *JSON* melalui *ConvertRecord*, baik dengan *delimiter semicolon* maupun *Excel/CSV* standar, sehingga data memiliki

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

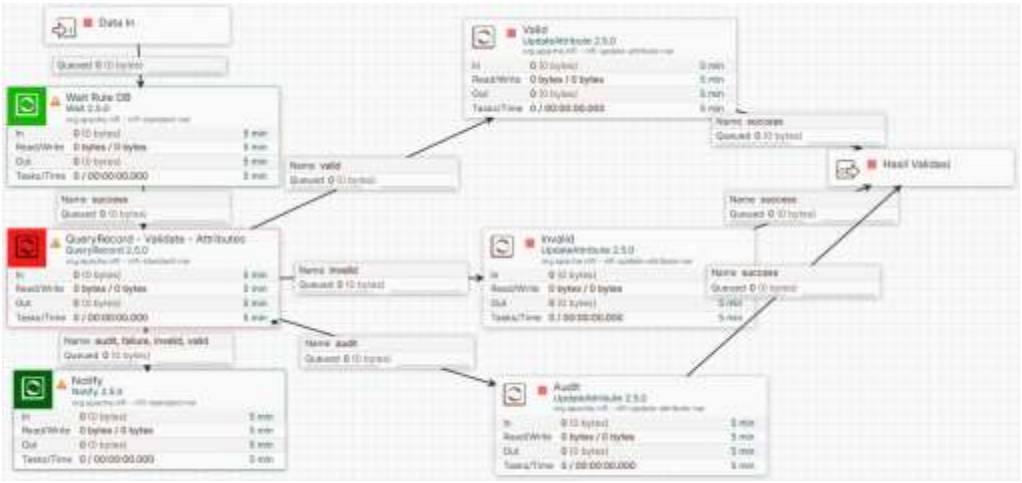
struktur seragam untuk proses validasi berikutnya. Setelah itu, data *JSON* disimpan sementara menggunakan *PutFile* dan *EvaluateJsonPath* untuk memastikan struktur *field* sesuai rule.

Tahap berikutnya adalah validasi rule menggunakan *QueryRecord* yang membandingkan atribut data dengan aturan yang berlaku. Data yang sesuai rule diberi label *Valid* melalui *UpdateAttribute* dan diteruskan ke *Hasil Validasi*. Sebaliknya, data yang tidak sesuai aturan diberi label *Invalid* dan dicatat menggunakan *Audit*. *ReplaceText* digunakan untuk membentuk query validasi, sedangkan *Notify* memberikan notifikasi status eksekusi rule. Dengan alur ini, *pipeline* mampu mengisolasi data valid dan tidak valid secara otomatis dari *spreadsheet*, memastikan hanya data terstandarisasi yang diteruskan ke tahap pemuatan selanjutnya, sebagaimana ditunjukkan pada Gambar 8.



Gambar 8. *Transform Spreadsheet Stage*

Alur pada gambar 9 menunjukkan bagaimana data dari *database diekstrak*, diberi konteks entitas, lalu divalidasi dengan *rule* yang telah ditentukan. Data masuk terlebih dahulu menunggu sinkronisasi rule di *Wait Rule DB*, kemudian jenis validasi ditentukan melalui RouteOnAttribute. Selanjutnya, processor *UpdateAttribute* digunakan untuk menetapkan tabel sumber seperti Mahasiswa, Dosen, Kelas, Mata Kuliah, dan Nilai. Setelah tabel ditentukan, *ExecuteSQLRecord* mengeksekusi query untuk mengambil data dari masing-masing tabel. Alur ini memastikan setiap entitas akademik diproses secara modular dan sesuai dengan konteksnya. Tahap berikutnya adalah validasi data. *Processor QueryRecord* digunakan untuk membandingkan atribut hasil ekstraksi dengan rule validasi. Data yang lolos validasi diberi label Valid dan diteruskan ke "Hasil Validasi", sementara data yang tidak memenuhi aturan diberi label Invalid dan dicatat menggunakan Audit. *Processor* tambahan seperti *ReplaceText* dan *LogAttribute* berfungsi mencatat *query* validasi yang dijalankan, sedangkan *Notify* memberikan notifikasi status eksekusi. Data yang sudah dipilah dapat disimpan menggunakan *PutFile* sebagai persiapan tahap *Load*.



Gambar 9. *Trasnform Database Pipeline*

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

3.2.3 *Load*

Tahap *Load* bertugas memuat data terstruktur yang telah tervalidasi dan ditransformasi ke dalam lingkungan penyimpanan Big Data yang skalabel, yaitu HDFS. Implementasi dilakukan dengan menggunakan *processor PutHDFS* (Gambar 10), yang secara otomatis menangani pemecahan dan pendistribusian data ke seluruh *cluster*. Data dimuat ke dalam cluster HDFS yang terdiri dari tiga *data node*, yang berfungsi memastikan ketersediaan tinggi (*High Availability*) dan toleransi kesalahan (*Fault Tolerance*).

Gambar 10. *Load Processor Pipeline*

Antarmuka pada Gambar 11 memperlihatkan hasil akhir proses *Load* di mana data valid dari *pipeline* ETL disimpan ke dalam HDFS. Setiap entitas data ditempatkan dalam direktori terpisah sesuai kategorinya, dengan konfigurasi *replication factor* yang memastikan ketersediaan tinggi dan toleransi kesalahan. Tampilan ini menjadi bukti bahwa data akademik yang telah tervalidasi berhasil diintegrasikan ke dalam penyimpanan terdistribusi, sehingga dapat digunakan sebagai *single source of truth* untuk kebutuhan analitik maupun pelaporan institusional. Selain itu, struktur penyimpanan yang rapi ini memudahkan proses audit dan akreditasi karena data telah dipisahkan berdasarkan entitas. Hasil ini juga menunjukkan kesiapan infrastruktur big data perguruan tinggi untuk mendukung pengelolaan data akademik dalam skala besar secara konsisten.

Gambar 11. Antarmuka HDFS

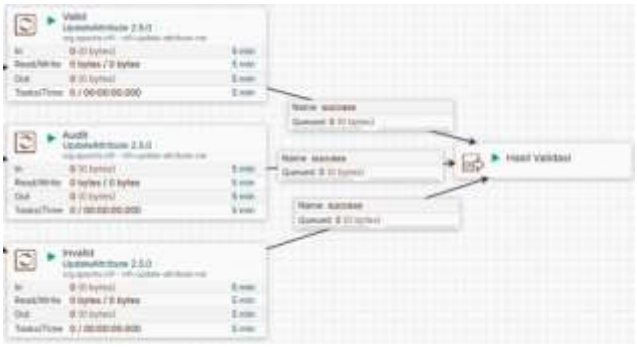
3.2.4 *Handling Invalid Data*

Data yang tidak lolos validasi tidak langsung dibuang, melainkan diarahkan ke direktori khusus *invalid data*. Mekanisme ini memberi kesempatan bagi operator untuk meninjau ulang data bermasalah dan melakukan koreksi

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

langsung pada sumber. Dengan demikian, pipeline berfungsi tidak hanya sebagai alat pemilah, tetapi juga sebagai mekanisme monitoring yang mendukung siklus perbaikan data secara berkelanjutan.

Gambar 12 memperlihatkan alur akhir penanganan data setelah tahap validasi. Processor *UpdateAttribute* digunakan untuk memberi label pada data menjadi tiga kategori: *Valid*, *Invalid*, dan *Audit*. Data yang lolos aturan diberi label *Valid* dan diteruskan ke *Hasil Validasi*. Data yang tidak sesuai aturan diberi label *Invalid* dan juga diarahkan ke *Hasil Validasi*, namun tetap disimpan terpisah sebagai catatan kesalahan. Sementara itu, jalur *Audit* mencatat metadata serta detail eksekusi proses, sehingga operator dapat melacak dan mengevaluasi kembali penyebab kegagalan validasi. Rancangan ini memastikan pipeline tidak hanya menghasilkan data tervalidasi, tetapi juga menyediakan sarana pengendalian mutu dan pelacakan kesalahan secara sistematis.



Gambar 12. Handling Invalid Data Pipeline

Mekanisme *invalid data handling* tidak hanya menjaga konsistensi data yang dimuat ke HDFS, tetapi juga berfungsi sebagai umpan balik strategis bagi perguruan tinggi dalam meningkatkan tata kelola data secara menyeluruh.

3.3 Uji Coba Hasil

Data tervalidasi divisualisasikan dalam sebuah *dashboard front-end*. *Dashboard* pada Gambar 13 ini menampilkan ringkasan data mahasiswa, dosen, mata kuliah, kelas, serta distribusi nilai, sehingga memudahkan pengguna dalam melakukan analisis. Uji coba dilakukan untuk mengevaluasi kinerja pipeline ETL yang telah dibangun. Fokus utama pengujian adalah mengukur persentase validitas data terhadap rule yang telah ditentukan pada tahap analisis. Setiap tabel data (mahasiswa, dosen, mata kuliah, kelas, dan nilai) diuji dengan cara menghitung jumlah *record* yang sesuai dengan aturan panjang *field*, nilai minimum dan maksimum, serta aturan *nullability*. Persentase validitas diperoleh dari perbandingan antara jumlah data yang lolos validasi dengan total keseluruhan data yang diuji.

Gambar 13. Antarmuka Detail Hasil Validasi

Tabel 8 menunjukkan hasil validasi pada PT1 untuk lima entitas data. Dari total 5.110 data Mahasiswa, sebanyak 4.923 (96,34%) dinyatakan valid, sementara 187 (3,66%) tidak valid. Untuk data Dosen, dari 452 baris, 441 (97,57%) valid dan 11 (2,43%) tidak valid. Pada entitas Mata Kuliah, validitas mencapai 98,60% dengan 704 data valid dari 714 total. Sementara itu, data Kelas menunjukkan 94,14% valid (1.251 dari 1.329 data). Entitas dengan tingkat kesalahan terbesar adalah Nilai, di mana dari total 18.632 data, 17.248 (92,57%) valid dan 1.384 (7,43%) tidak valid. Hasil ini

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

mengindikasikan bahwa pipeline berhasil mendeteksi kelemahan terutama pada pengisian nilai, misalnya format nilai angka di luar batas atau ketidaklengkapan field wajib.

Tabel 8. Hasil Uji Coba Data Perguruan Tinggi 1

Data	ΣBaris	Valid:Invalid Baris (%)	ΣKolom	Valid:Invalid Kolom(%)
Mahasiswa	1.369	0:100	36.963	93.9:6.1
Dosen	108	0:100	1.944	93.5:6.5
Mata Kuliah	313	100:0	5.321	100:0
Kelas	677	100:0	7.447	100:0
Nilai	1.850	100:0	9.250	100:0

Tabel 9 memperlihatkan hasil validasi pada PT 2 yang secara umum lebih baik dibanding PT 1. Dari 4.367 data Mahasiswa, 4.306 (98,60%) valid dan hanya 61 (1,40%) tidak valid. Pada data Dosen, validitas mencapai 98,07% dengan 408 dari 416 data valid. Untuk Mata Kuliah, 685 dari 692 data valid (98,99%**).** Data Kelas juga relatif konsisten, dengan 1.118 dari 1.150 data valid (97,22%)**.** Entitas Nilai menunjukkan validitas yang tinggi yaitu 94,33% (15.928 dari 16.882 data), lebih baik dibanding PT1. Perbedaan ini menegaskan adanya variasi kualitas input antar perguruan tinggi yang dapat dipengaruhi oleh sistem informasi akademik maupun prosedur pengelolaan data. Pipeline mampu mengungkap perbedaan tersebut secara kuantitatif dan sistematis.

Tabel 9. Hasil Uji Coba Data Perguruan Tinggi 2

Data	ΣBaris	Valid:Invalid Baris (%)	ΣKolom	Valid:Invalid Kolom(%)
Mahasiswa	1.000	20.9:79.1	26.000	91.1:8.9
Dosen	92	25:75	1.104	86.1:13.9
Mata Kuliah	1.000	84.1:5.9	16.000	99:1
Kelas	1.000	99.4:0.6	8.000	99.9:0.1
Nilai	1.000	93.7:6.3	5.000	98.7:1.3

Hasil uji coba menunjukkan *pipeline* ETL yang dibangun dapat mengekstrak data dari *spreadsheet* dan database dengan tingkat keberhasilan 100%. Pada tahap transformasi, sekitar 8% data awal terdeteksi error (misalnya field kosong atau panjang field tidak sesuai aturan). Data yang telah tervalidasi dimuat ke dalam HDFS *cluster* tiga node dengan distribusi merata, mendukung ketersediaan dan toleransi kesalahan. *Dashboard* visualisasi menampilkan data secara *real time* dan memudahkan pengguna dalam memantau kualitas data.

Pipeline ETL berbasis *Apache NiFi* dan HDFS terbukti efektif dalam mengatasi permasalahan integrasi dan validasi data perguruan tinggi. Proses ekstraksi otomatis dari dua sumber berbeda mempercepat integrasi dibandingkan metode manual. Validasi berbasis aturan memastikan data sesuai standar, mengurangi kesalahan input, dan meningkatkan kualitas informasi. Penyimpanan terdistribusi di *HDFS* menyediakan skalabilitas untuk pertumbuhan data serta toleransi kesalahan pada node. Dibandingkan metode tradisional, *pipeline* ini mampu meningkatkan konsistensi data sekaligus mengurangi beban kerja operator.

Hasil uji coba ini memberikan gambaran sejauh mana *pipeline* ETL mampu menjalankan fungsi mengukur validatas data sesuai rule yang telah dirancang. Dengan demikian, tahapan ini tidak hanya berfungsi sebagai verifikasi teknis, tetapi juga sebagai indikator kualitas data setelah melalui proses ETL berbasis big data.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan pipeline ETL big data untuk data terstruktur di perguruan tinggi. Pipeline mampu: mengekstraksi otomatis data dari spreadsheet dan basis data, melakukan transformasi berbasis rule, menyimpan data tervalidasi ke HDFS dengan high availability dan fault tolerance. Hasil uji coba menunjukkan pipeline dapat menilai validitas data dengan akurat, menyoroti kelemahan kualitas input data, dan menyediakan penyimpanan data terintegrasi yang skalabel. Implementasi ini dapat menjadi fondasi sistem tata kelola data akademik di perguruan tinggi serta mendukung audit, akreditasi, dan evaluasi berbasis data. Keterbatasan penelitian ini adalah belum mencakup integrasi data tidak terstruktur. Pada penelitian lanjutan, pipeline dapat diperluas dengan Apache Spark untuk analisis prediktif serta integrasi data semi-terstruktur seperti log aktivitas mahasiswa.

JURNAL SISTEM INFORMASI TGD
Volume 5, Nomor 2, Maret 2026, Hal 481-492
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
https://ojs.trigunadharma.ac.id/index.php/jsi

UCAPAN TERIMAKASIH

Penulis menyampaikan terima kasih kepada Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek) melalui Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi yang telah memberikan dukungan pendanaan penelitian ini dengan nomor kontrak 123/C3/DT.05.00/PL/2025. Selain itu, penghargaan diberikan kepada berbagai pihak yang turut membantu dalam aspek teknis, konseptual, maupun administratif sehingga penelitian ini dapat terlaksana dengan baik.

DAFTAR PUSTAKA

[1] M. I. Baig, L. Shuib, dan E. Yadegaridehkordi, "Big data in education: a state of the art, limitations, and future research directions," *Int. J. Educ. Technol. High. Educ.*, vol. 17, no. 1, hal. 44, Des 2020, doi: 10.1186/s41239-020-00223-0.

[2] A. R. Munappy, J. Bosch, dan H. H. Olsson, "Data Pipeline Management in Practice: Challenges and Opportunities," 2020, hal. 168–184. doi: 10.1007/978-3-030-64148-1_11.

[3] M. Y. Amare dan S. Simonova, "Learning analytics for higher education: proposal of big data ingestion architecture," *SHS Web Conf.*, vol. 92, hal. 02002, Jan 2021, doi: 10.1051/shsconf/20219202002.

[4] L. G. Tanasescu, A. Vines, A. R. Bologa, dan C. A. Vaida, "Big Data ETL Process and Its Impact on Text Mining Analysis for Employees' Reviews," *Appl. Sci.*, vol. 12, no. 15, hal. 7509, Jul 2022, doi: 10.3390/app12157509.

[5] H. Foidl, V. Golendukhina, R. Ramler, dan M. Felderer, "Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers," *J. Syst. Softw.*, vol. 207, hal. 111855, Jan 2024, doi: 10.1016/j.jss.2023.111855.

[6] D. Tosi, R. Kokaj, dan M. Roccetti, "15 years of Big Data: a systematic literature review," *J. Big Data*, vol. 11, no. 1, hal. 73, Mei 2024, doi: 10.1186/s40537-024-00914-9.

[7] Z. Shojaee Rad dan M. Ghobaei-Arani, "Data pipeline approaches in serverless computing: a taxonomy, review, and research trends," *J. Big Data*, vol. 11, no. 1, hal. 82, Jun 2024, doi: 10.1186/s40537-024-00939-0.

[8] N. Setiyawati, D. H. Bangkalang, dan G. W. Asmara, "Design and Implementation of an ETL Pipeline for Prospective Student Data Analysis in Higher Education Admissions," *SISTEMASI*, vol. 14, no. 5, hal. 2125, Sep 2025, doi: 10.32520/stmsi.v14i4.5158.

[9] N.-S. Chen, C. Yin, P. Isaias, dan J. Psotka, "Educational big data: extracting meaning from data for smart education," *Interact. Learn. Environ.*, vol. 28, no. 2, hal. 142–147, Feb 2020, doi: 10.1080/10494820.2019.1635395.

[10] H. K. Israel Mnsen, B. Purnomosidi, R. Kartadie, dan D. Kurnaedi, "DATA PIPELINE ARCHITECTURE FOR ACADEMIC INFORMATION SYSTEM AT AKADEMI TEKNIK BIAK," *J. Intell. Softw. Syst.*, vol. 3, no. 1, hal. 1, Jul 2024, doi: 10.26798/jiss.v3i1.1335.

[11] D. Chanda, "Automated ETL Pipelines for Modern Data Warehousing: Architectures, Challenges, and Emerging Solutions," *Eastasouth J. Inf. Syst. Comput. Sci.*, vol. 1, no. 03, hal. 209–212, Apr 2024, doi: 10.58812/esiscs.v1i03.523.

[12] A. Budi Trisnawan, "Pemanfaatan Big Data dalam Sistem Informasi untuk Pengambilan Keputusan Strategis," *J. Inf. Syst. Educ. Dev.*, vol. 3, no. 3, hal. 39–43, Sep 2025, doi: 10.62386/jised.v3i3.163.

[13] S. Kumar Singu, "ETL Process Automation: Tools and Techniques", doi: 10.56472/25832646/JETA-V2I1P110.

[14] B. Johnson Mary *et al.*, "Managing Data Quality and Consistency in Real-Time ETL for Streaming Applications: A Comparative Analysis of Modern ETL Frameworks," 2025. [Daring]. Tersedia pada: https://www.researchgate.net/publication/392589655

[15] Gayatri Tavva, "Maximizing ETL efficiency: Patterns for high-volume data," *Int. J. Sci. Res. Arch.*, vol. 15, no. 2, hal. 1063–1070, Mei 2025, doi: 10.30574/ijsra.2025.15.2.1477.