JURNAL RISET INFORMATIKA **P-ISSN: 2656-1743 |E-ISSN: 2656-1735**
Vol. 7, No. 4. September 2025 DOI: https://doi.org/10.34288/jri.v7i4.395
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Integration of OCR Technology with ETL Processes for Automating Data Pipeline of Financial Disbursement Documents at BPS Sukabumi Regency

Muhammad Raihan Izharul Haq[-1], Gina Purnama Insany[-2], Somantri[-3]

Informatics Engineering
Nusa Putra University
muhammad.raihan_ti21@nusaputra.ac.id[-1], gina.purnama@nusaputra.ac.id[-2], somantri@nusaputra.ac.id[-3]

Abstract

In the digital era, managing archival data poses challenges for many institutions, including Badan Pusat Statistik (BPS) of Sukabumi Regency, especially when dealing with unstructured PDF documents. This study develops a data pipeline by effectively integrating Optical Character Recognition (OCR) technology with Extract, Transform, Load (ETL) processes. Unstructured data from financial disbursement documents, such as SPM and SP2D, were automatically extracted with high accuracy, achieving an average of 98.52% for SPM using a combination of OCR and PDFPlumber, and 100% for SP2D extracted using PDFPlumber. Extraction results were stored in a data warehouse, then transformed using Apache Spark and loaded into data marts. ETL process was automated using Apache Airflow, which operated reliably according to dependencies. The processed data were presented through an interactive Looker Studio dashboard in real-time, supporting efficient archive management and more informed decision-making. This study not only provides a solution to existing archival management problems but also opens opportunities for further development in the application of big data technologies and business process automation in public sector.

Keywords: Big Data, Optical Character Recognition (OCR), Extract Transform Load (ETL), Automated Data Pipeline, Financial Disbursement

Abstrak

Di era digital, pengelolaan data arsip menjadi tantangan bagi banyak institusi, termasuk Badan Pusat Statistik (BPS) Kabupaten Sukabumi, khususnya dalam menangani dokumen PDF yang tidak terstruktur. Penelitian ini mengembangkan sebuah data pipeline dengan mengintegrasikan teknologi Optical Character Recognition (OCR) secara efektif dengan proses Extract, Transform, Load (ETL). Data tidak terstruktur dari dokumen pencairan keuangan, seperti SPM dan SP2D, diekstraksi secara otomatis dengan akurasi tinggi, yaitu rata-rata 98,52% untuk SPM dengan kombinasi OCR dan PDFPlumber, serta 100% untuk SP2D menggunakan PDFPlumber. Hasil ekstraksi disimpan dalam data warehouse, kemudian ditransformasi menggunakan Apache Spark dan dimuat ke dalam data marts. Proses ETL diotomatisasi menggunakan Apache Airflow yang berjalan andal sesuai dengan ketergantungannya. Data yang telah diproses disajikan melalui dasbor interaktif di Looker Studio secara real-time, mendukung pengelolaan arsip yang efisien serta pengambilan keputusan yang lebih informasional. Studi ini tidak hanya memberikan solusi atas permasalahan pengelolaan arsip yang ada, tetapi juga membuka peluang pengembangan lebih lanjut dalam penerapan teknologi big data dan otomasi proses bisnis di sektor publik.

Kata kunci: Big Data, Optical Character Recognition (OCR), Extract Transform Load (ETL), Data Pipeline Otomatis, Pencairan Keuangan

INTRODUCTION

Archival management plays a vital role in supporting performance and operational efficiency of an institution (Jamaluddin, Nurfadila, & Isgunandar, 2023). Well-organized records facilitate easier access to information and support decision-making processes, ensuring both short-term functionality and long-term institutional continuity (Darmansah, Agung, Hasbih, & Lucky Tirta, 2023). At BPS Sukabumi Regency, financial disbursement records such as *Surat Perintah Membayar* (SPM) and *Surat Perintah Pencairan Dana* (SP2D) are central to financial administration

(Wulandari & Maula, 2022). SPM is a document containing instruction to a treasurer to process payment for a transaction while SP2D is issued by *Kantor Pelayanan Perbendaharaan Negara* (KPPN) as an official basis for fund disbursement from state treasury to recipient's account (Bakari, Karamoy, & Lambey, 2022).

As data volumes and complexity continue to increase, there is a growing need for more efficient data processing solutions (Li, Chen, & Shang, 2022). Within the big data paradigm, challenges primarily stem from volume, velocity, and variety of data types (Siti Sarah Sobariah Lestari, Gina Purnama Insany, Dede Sukmawan, & Faiz Dzulfikar Yusuf, 2023). Notably, the global market for big data and data engineering services is projected to reach $75.55 billion in 2024, surging to $169.9 billion by 2029 with a 17.6% Compound Annual Growth Rate (CAGR) (Yamjala, 2024). Big Data also plays a critical role in development of Artificial Intelligence (AI) systems, with its integration shown to significantly enhance decision-making processes, drive product innovation, and improve operational efficiency across various sectors (Zulfadli & Syahputra, 2024).

Currently, document management processes at BPS Sukabumi remain predominantly manual. Such practices are often time-consuming, susceptible to human error, and ultimately hinder overall organizational productivity. Without proper automation in extracting and processing archival data, timely access to crucial information becomes difficult, thereby causing delays in analysis and decision-making.

Optical Character Recognition (OCR) is a key enabling technology that extracts textual information from scanned documents (Irimia, Harbuzariu, Hazi, & Iftene, 2022). When combined with an automated Extract, Transform, Load (ETL) data pipeline, it not only improves processing efficiency but also allows for deeper and more meaningful data analysis (Murtiwiyati, Hansel, & Leli, 2024).

This study proposes an integrated system utilizing OCR and PDFPlumber for extracting data from PDF documents, followed by transformation using Python, and structured storage in a PostgreSQL data warehouse. Fundamentally, a data warehouse serves as a repository for organizational historical data, focusing on aggregated information rather than highly detailed operational data (Atsila Imanda et al., 2024). In practice, data engineers hold significant responsibility for managing large-scale data processes, which include data cleaning, transformation, and distribution, to ensure that business decision-making is based on reliable, consistent, and timely accessible data (Riza, Aulia, Kolin, & Mustaqim, 2024). By leveraging the data warehouse, historical data can be analyzed comprehensively to yield accurate and consistent insights, supporting more effective decision-making (Adrezo & Ermatita, 2023), and a well-designed data warehouse architecture positively impacts quality and precision of business decisions (Fauzi, Noor, Ardyansyah, & Semesta, 2023).

Data processing and aggregation are performed using Apache Spark. Recent studies have shown that Apache Spark consistently delivers strong performance, particularly when handling complex queries on large datasets and leveraging multi-core processors efficiently (Arjan Rangkuti, Zihni Athallah, Harwani Barus, Afisa Rani, & Ikhsan Setiawan, 2023). Apache Spark also has demonstrated an average speed that is approximately 4.99 times faster than Hadoop MapReduce, making it a highly efficient choice for large-scale data workflows (Wibawa, Wirawan, Mustikasari, & Anggraeni, 2022).

The entire ETL workflow is managed using Apache Airflow, applying scheduled batch processing (Sanchez, 2022). Implementation of Directed Acyclic Graphs (DAG) in Airflow enables users to manage task execution in parallel or sequentially, improving efficiency and reliability in large-scale data processing (Wahyudi et al., 2022). A comparative analysis highlights Apache Airflow's superior performance compared to conventional ETL tools, particularly in setup simplicity, operational efficiency, scalability, error management, and integration flexibility (Eeti, GOEL, & KUSHWAHA, 2022). To further support deployment consistency and scalability, Docker is utilized as a containerization platform. This approach improves application lifecycle management, allowing developers to easily build, test, and deploy applications across diverse environments without concerns over system configuration differences (Subekti et al., 2024).

The objective of this study is to develop an automated data processing system that streamlines the management of financial documents and supports timely, data-driven decision-making. The proposed system integrates OCR with an ETL-based architecture to automate the extraction and transformation of unstructured financial disbursement documents at BPS Sukabumi Regency.

RESEARCH METHODS

This study employed a mixed methods approach, combining both qualitative and

quantitative techniques. Qualitative methods were applied during data collection to gain a deep understanding of business requirements and formulate system specifications. This involved conducting interviews and direct observations within working environment of BPS Sukabumi Regency. Meanwhile, quantitative techniques were used to evaluate system performance, particularly in assessing accuracy of OCR technology in processing unstructured data such as scanned documents in PDF format.

For system development, this research adopted the Analysis, Design, Development, Implementation, and Evaluation (ADDIE) model, a systematic and iterative framework consisting of five main phases: needs analysis, solution design, system development, implementation, and evaluation (Syuhada, Hidayat, Mulyati, & Giri Persada, 2023). This model was selected because it aligns with the objectives of this study, which is to design and develop an integrated prototype system to solve unstructured document processing issues related to financial disbursement activities at BPS Sukabumi Regency.

A. Analysis

In this phase, a qualitative approach was employed to gather data related to issues in managing unstructured financial disbursement archives at BPS Sukabumi Regency. Data collection was conducted through several methods, including observation, interviews, and literature review. The primary focus of this analysis stage was to identify key problems, such as manual handling of PDF documents, and to define system requirements for the solution to be developed. A crucial aspect of this phase involved ensuring that both software and hardware components were properly identified and fulfilled. Once these essential elements were in place, the system could be reliably tested and expected to operate efficiently and stably within its intended environment (Kamdan, Somantri, Sundayana, & Kharisma, 2023).

Table 1. Hardware Requirement

No.	Requirement	Function
1.	Prosesor AMD Ryzen 7 5700U with Radeon Graphics 1.80 GHz	Handles system and application computing processes efficiently
2.	24 GB RAM	A minimum of 16 GB RAM is recommended to ensure optimal performance
3.	OS 64-bit, x64-based processor	Ensures compatibility with software used in the system

Table 2. Software Requirement

No.	Requirement	Function
1	Docker Desktop	Provides a containerized environment to run system services
2.	Astronomer	Manages local Airflow projects
3.	Apache Spark (within Dockerfile)	Enables parallel and distributed processing of large datasets within containers
4.	Pytesseract (within Dockerfile)	Extract text from scanned PDF file
5.	PostgreSQL	Serves as a data warehouse system to store, organize, and manage large volumes of structured data
6.	Spreadsheet	Used as a temporary data mart
7.	Looker Studio	Delivers data in the form of interactive dashboards for visualization
8.	Google Cloud Platform (GCP) Account	Manages access and authorization using IAM to enable system connectivity with Google Sheets.

B. Design

The planning and architectural design of a data pipeline are critical stages in data management. These steps define how data will be collected, processed, stored, and distributed efficiently. A well-structured architecture ensures a smooth flow of data, maintains data integrity and quality, and enables accurate and timely analysis to support effective decision-making (Suriansyah, Mz, Rachman, & Pratiwi, 2025).

This phase involves designing the data pipeline architecture for ETL process, which consists of two main components. The first handles ETL from data source to data warehouse, while the

second covers ETL from data warehouse to data marts. The pipeline is designed to ensure data is orchestrated efficiently and maintained in a coherent, scalable structure. Orchestration is performed using Apache Airflow, deployed on Astronomer platform. It comprises three main DAGs: SPM DAG, SP2D DAG, and Load to Marts DAG. This phase also includes developing data models for a PostgreSQL-based data warehouse. Data transformation is conducted using Apache Spark prior to loading into data marts, which are implemented using spreadsheet-based storage. Processed data is then visualized through interactive dashboards created with Looker Studio.

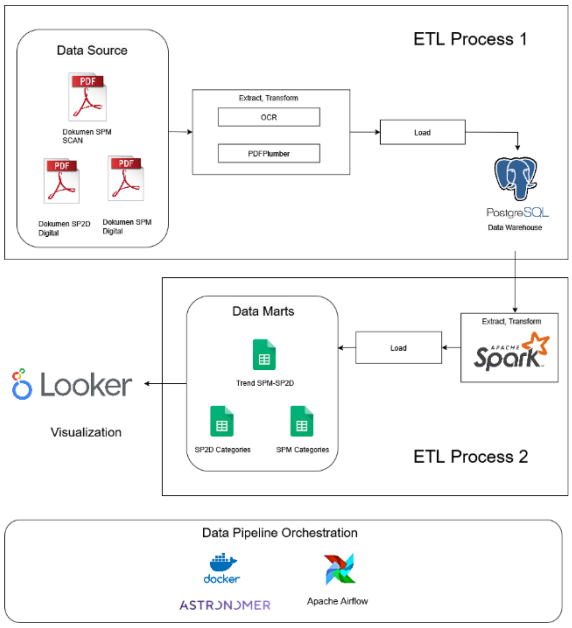

Figure 1. Data Pipeline Architecture Diagram

- ETL Process 1

This process begins by ingesting two types of PDF files: SPM in both scanned and digital formats, and SP2D in digital format only. Scanned SPM files are processed using OCR, while digital PDF files for both SPM and SP2D are extracted using PDFPlumber. The extracted data is then transformed into structured JSON files, with one JSON file generated for each SPM and SP2D file. Finally, the structured data is loaded into a PostgreSQL-based data warehouse for storage and further analysis.

The process of extracting and transforming SPM data:
1. The process begins by opening a PDF file that needs to be processed, whether it's a digitally generated text PDF or a scanned image PDF.
2. The system determines whether the PDF contains selectable digital text or scanned images requiring OCR.
3. If the PDF is scanned, pages are rendered as images and processed using OCR, including pre-processing, cropping, and text detection.
4. If the PDF contains digital text, content is extracted directly without the need for OCR or image rendering.
5. SPM data is cleaned and required information is extracted from the text.
6. Extracted SPM data is added to a dictionary and appended to a list.
7. A check is performed to determine whether another PDF file exists in the folder; if so, the next file is opened and processed.
8. Once all files are processed, the list of dictionaries is written to a JSON file.
9. The process concludes after saving the data to the JSON file.

The process of transforming and loading SPM data:
1. The process begins by attempting to establish a connection to data warehouse.
2. If the connection is successful, the system proceeds to read JSON file containing SPM data.
3. Each record is validated to ensure required fields are not missing and SPM entry does not already exist.
4. Invalid or duplicate records are skipped.
5. Valid records proceed to insertion process, where date information is added to dim_tanggal table, and SPM data is inserted into fact_spm table.
6. The process repeats until all records in JSON file have been processed.

The process of extracting and transforming SP2D data:
1. The process begins by opening and reading digital PDF files to extract tables.
2. Each row in the extracted tables is iterated for further processing.
3. If a row is identified as the first row on the first page, it is skipped to avoid redundant header data.
4. The system verifies whether the row still contains valid data. Rows without data are skipped.
5. Valid rows undergo a data cleaning and extraction procedure.
6. Cleaned data is appended to a list of dictionaries.
7. Upon completing file processing, the file name and corresponding data are stored in the dictionary list.

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v7i4.395
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

8. Accumulated data is sorted based on the SP2D date.
9. If additional PDF files are present in the folder, the process repeats for each remaining file.
10. After all files are processed, data is sorted again based on month and year extracted from file names.
11. Finally, structured data is saved into a JSON file.

The process of transforming and loading SP2D data:
1. The process begins by attempting to establish a connection to data warehouse.
2. If the connection is successful, the system reads a JSON file containing SP2D data.
3. Each record undergoes validation to check for missing important fields and to ensure the SP2D entry does not already exist.
4. Valid records are inserted into the dim_tanggal table for date information and into the fact_sp2d table for SP2D data.
5. The system also checks whether an SP2D record already exists.
6. The process repeats until all records in the JSON file are processed, and then process ends.

- ETL Process 2

In the second ETL process, existing SP2D and SPM data stored in a data warehouse are extracted and transformed using Apache Spark, then loaded into spreadsheets serving as data marts. These data marts include several analytical views. The Trend SP2D SPM data mart presents a time-based comparison between SP2D and SPM values across different periods. The SP2D Categories data mart provides aggregated SP2D data based on SP2D and SPM types, grouped by time dimensions. It is important to note that the SPM type attribute in this data mart refers to the type of SPM document that serves as the basis for issuing SP2D, which may differ from the classification used in the SPM Categories data mart, as both originate from different process stages and data sources. The SPM Categories data mart presents aggregated SPM data categorized by SPM type and organized by time dimensions.

The process of extracting and transforming data in ETL 2:
1. The process commences by attempting to establish a connection to data warehouse.
2. If the connection to data warehouse is successful, the table is read using Spark SQL.
3. Subsequently, the read data is stored in partitions.
4. Process concludes after data has been successfully stored in partitions.

The process of transforming and loading data in ETL 2:
1. The process begins by attempting to establish a connection to data marts.
2. If the connection is successful, the process enters a loop that iterates over relevant sheets.
3. In each iteration, it starts by reading partitioned data in Parquet format.
4. Raw data is then converted into the required format or structure for further processing.
5. The corresponding sheet is opened to allow data update operations.
6. Before writing new data, sheet contents are cleared to prevent duplication or inconsistency.
7. The sheet is then updated with newly converted data.
8. The system checks whether the looping process for all target sheets is complete. If not, it continues to the next sheet. Otherwise, the operation concludes.

- Data Orchestration

The orchestration of this weekly scheduled process is designed to automatically execute a series of ETL stages using a DAG-based approach. The workflow involves integration and transformation of data from two primary sources, namely SPM and SP2D, through an initial ETL process into data warehouse, followed by a second ETL process that loads the refined data into data marts.

The SPM DAG for ETL 1 begins with the extract-transform stage, which retrieves and initially transforms data from the source. This is followed by the transform-load stage, where further transformations are applied and data is loaded into the warehouse. Once the ETL 1 process for SPM is complete, the system automatically triggers the next step by running the DAG for ETL 2.

Figure 2. SPM DAG Flowchart

The SP2D DAG for ETL 1 begins with the extract-transform stage, which involves retrieving data from the source and performing initial transformations. This is followed by the transform-load stage, where data undergoes further processing and is loaded into the data warehouse. Once the ETL 1 process for SP2D is complete, the system automatically triggers the next step by executing the DAG for ETL 2 as a continuation of the data pipeline.

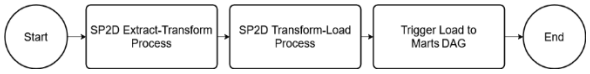

Figure 3. SP2D DAG Flowchart

The Load to Marts DAG for the ETL 2 process is automatically triggered upon completion of the ETL 1 DAG, applicable to both SPM and SP2D cases. This stage begins with the extract-transform phase, which retrieves and processes data from the data warehouse. The transformed data then moves to the transform-load phase, where it is loaded into data marts as the final storage destination. These data marts are then ready to be connected to Looker Studio dashboards for reporting and analysis.

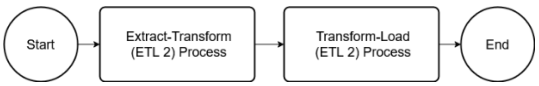

Figure 4. Load to Marts DAG Flowchart

C. Development

The system development is carried out within a containerized environment using Docker, ensuring consistency across different platforms. Data orchestration is managed through the Astronomer platform for Apache Airflow. Each component of the data pipeline, from extraction and transformation to loading, is developed and tested within this environment to ensure optimal system integration.

D. Implementation

The system is implemented within the working environment of BPS Sukabumi Regency. By utilizing Docker, the system that has been developed and tested in the development environment can be consistently deployed on other machines, including user computers at BPS, without the need for extensive reconfiguration.

E. Evaluation

Evaluation was conducted iteratively and in parallel throughout system development process to ensure that each stage of the ETL workflow operated as intended. Testing was performed at every phase, from design stage through to implementation, allowing for early and continuous improvements. The testing methods included black-box testing to assess the overall functionality of the system (Yusuf Alfiansyah & Arisandi, 2023), as well as quantitative evaluation of data extraction model's performance. This was done by comparing the extracted results with original document content using Python's SequenceMatcher library to calculate the similarity score as a percentage.

RESULTS AND DISCUSSION

This section presents system implementation and testing results. The evaluation focuses on system testing, including black-box testing and accuracy assessment of the data extraction model.

A. Implementation

- Data Warehouse

The development of PostgreSQL-based data warehouse was carried out using DBeaver, in accordance with previously designed data modeling schema. This data warehouse is intended to store and manage SPM and SP2D data in a structured manner.

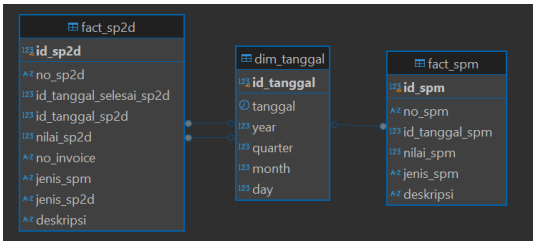

Figure 5. Entitiy Relationship Diagram (ERD) Data Warehouse

- ETL Pipeline Orchestration System

Before running the system, Docker Desktop must be activated until its status shows "Engine running," indicating that the service is operational. Once Docker is active, navigate to the project directory. Start the system via the command line using astro dev start.

When the system is running, Apache Airflow will also launch automatically. Its status can be verified by accessing the main interface through a web browser. This interface displays a list of DAGs, execution statuses, and provides navigation to other available features.

After successfully accessing the Airflow web UI, configure the required connections for pipeline execution, including connections to the PostgreSQL data warehouse and spreadsheet-based data marts.

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v7i4.395
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

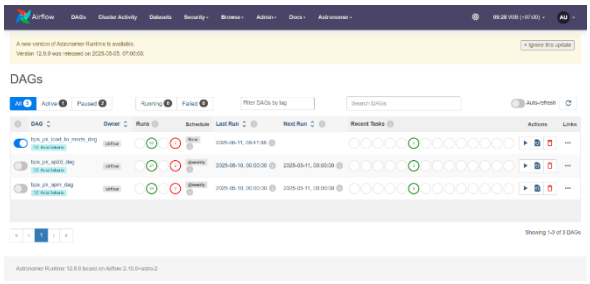

Figure 6. Airflow Main Page

- Data Marts

Upon completion of the second ETL process, processed SPM and SP2D data in the data warehouse are exported to data marts in spreadsheet format. This file contains three sheets: Trend_Data, SPM_Categories, and SP2D_Categories, each presenting transformed data intended for further analysis.

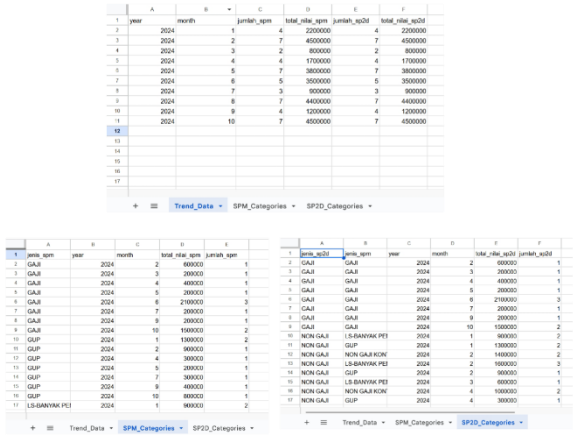

Figure 7. Data Marts

- Data Visualization

Data contained in the spreadsheet generated from data marts is subsequently integrated into *Pencairan Keuangan* (PK) Report dashboard in Looker Studio for data visualization. Each sheet is directly linked to its corresponding page on the dashboard, namely PK Overtime, SP2D Categories, and SPM Categories, facilitating easy monitoring and interactive visual analysis.

Figure 8. Looker Studio Dashboard

B. Tests Results

- Black-box Testing

Black-box testing was conducted to verify that system features function as intended and align with defined specifications. This testing focused on validating whether each function produces the expected output for given inputs, ensuring the system behaves as expected from a user's perspective, without regard to internal implementation.

Table 3. Black Box Test for ETL Process 1

Test Case	Expected Output	Desciption
A valid digital SPM PDF file	SPM data is successfully extracted from the PDF file	Passed
A valid scanned SPM PDF file	SPM data is successfully extracted from the PDF file	Passed
An invalid digital SPM PDF file	SPM data is not extracted and the extracted value is returned as null	Passed
An invalid scanned PDF file	SPM data is not extracted and the extracted value is returned as null	Passed
Complete and valid SPM data not yet present in the data warehouse	SPM data is inserted into dim_tanggal and fact_spm tables	Passed

A PDF file with a valid table and complete SP2D data	SP2D rows are cleanly extracted and sorted	Passed
A PDF file without a valid table or with incorrect format	No data extraction is performed	Passed
Complete and valid SP2D data not yet present in the data warehouse	SP2D data is inserted into dim_table and fact_sp2d tables	Passed

Table 4. Black Box Test for ETL Process 2

Test Case	Expected Output	Desciption
Data from dim_tanggal, fact_spm, and fact_sp2d tables is read using Spark SQL; the output is saved in spark_output directory with appropriate partitioning	Parquet files are successfully written to the spark_output directory with correct partition structure	Passed
Data is sent to the Trend_Data, SP2D_Categories, and SPM_Categories sheets in the target spreadsheet	Data is successfully transmitted and appears in the correct sheets within the specified spreadsheet	Passed

Table 5. Black Box Test for Data Orchestration

Test Case	Expected Output	Desciption
SPM DAG tasks are triggered and executed	SPM DAG tasks are successfully executed in the correct sequence	Passed
SPM DAG is configured to run once per week	SPM DAG is successfully scheduled to run on a weekly basis	Passed
SP2D DAG tasks are triggered and executed	SP2D DAG tasks are successfully executed in the correct sequence	Passed
SP2D DAG is configured to run once per week	SP2D DAG is successfully scheduled to run on a weekly basis	Passed
Data load to marts DAG is triggered and executed	Data load to marts DAG is successfully triggered and executed in sequence	Passed

- Data Extraction Model Accuracy Testing

To evaluate the model's performance in extracting information from SPM documents in PDF format, both scanned and digitally generated, a series of tests were conducted by comparing the extracted data with manually entered reference data, also known as ground truth. SP2D documents, available only in digital PDF format, were excluded from this accuracy assessment, as their extraction consistently produced outputs that were 100% identical to the reference data. Given PDFPlumber's high precision in extracting text from digital documents, a separate accuracy test for SP2D was deemed unnecessary.

Evaluation employed a string similarity approach using the SequenceMatcher algorithm from Python's difflib library to measure how closely extracted values matched original data. Similarity scores ranged from 0 to 1, with 1 indicating a perfect match. Assessment focused on calculating overall average similarity across five primary fields: no_spm, tanggal_spm, nilai_spm, jenis_spm, and deskripsi. In addition to average score, percentage of exact matches was recorded to reflect how often extracted data was completely identical to ground truth.

A total of five JSON files were used in the evaluation. One file contains forty-eight extracted entries, while the other four files each contain fifty entries:

Table 6. Evaluation of SPM Data Extraction Model

Number of Files	Processing Time	Overall Avg. Similarity
48	±18 minutes	98.59%

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v7i4.395
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

50	±30 minutes	97.07%
50	±22 minutes	98.09%
50	±24 minutes	99.34%
50	±23 minutes	99.51%

Evaluation results indicate that the extraction model demonstrates exceptionally high accuracy, with an overall average similarity rate of 98.52%. Fields with structured and numerical content, such as nilai_spm, tanggal_spm, and no_spm, consistently achieved high similarity levels and exact matches. In contrast, the deskripsi field, which contains free-form text, showed greater variability, although it remained within an acceptable similarity range.

In terms of processing time, digital PDF files were processed more quickly compared to scanned PDF files. For example, in the first JSON file, which contained five out of six documents in digital PDF format, the extraction process took approximately 18 minutes, which was faster than for files with more scanned PDFs.

Furthermore, for scanned documents, processing time can become significantly longer when the page containing SPM data later in the document, such as on page 16 instead of page 4. This delay is due to additional time needed to locate the relevant page containing SPM information.

The extraction process relies heavily on identifying pages containing phrases such as "SURAT PERMINTAAN PEMBAYARAN" as indicators of SPM data. However, if the page is blurry, damaged, partially cut off, or difficult to read due to factors such as low scan quality or visual noise, detection and extraction may be severely disrupted. Under such conditions, the system may fail to recognize the target page, hindering its ability to extract required information automatically and accurately.

CONCLUSIONS AND SUGGESTIONS

Conclusion

This study successfully integrated OCR into the ETL process to manage financial disbursement documents at BPS Sukabumi Regency, covering both scanned and digital formats. Unstructured data from SPM and SP2D were accurately extracted into structured form, loaded into a data warehouse, and transformed into data marts using Apache Spark. The extraction model achieved a high average similarity of 98.52%, with strong accuracy in structured fields and acceptable results in descriptive ones. The ETL pipeline was fully automated using Apache Airflow, running reliably through scheduled DAGs. Final outputs were visualized in real time through an interactive Looker Studio dashboard, enhancing data monitoring and supporting efficient decision-making.

Suggestion

This study has successfully developed an automated data pipeline system for managing financial disbursement archival documents. However, further improvements are necessary in terms of infrastructure and scalability to support deployment in large-scale production environments. It is recommended to migrate the system to cloud computing platforms such as AWS, Google Cloud Platform (GCP), or Microsoft Azure to ensure better scalability, reliability, and maintainability.

REFERENCES

Adrezo, M., & Ermatita, E. (2023). Implementasi Pentaho Pada Perancangan Data Warehouse Perusahaan Jasa Pengiriman (PT. Tiki Palembang). *Jurnal Teknik Informatika Dan Sistem Informasi*, *10*(2), 2407–4322. Retrieved from http://jurnal.mdp.ac.id

Arjan Rangkuti, P., Zihni Athallah, M., Harwani Barus, T., Afisa Rani, N., & Ikhsan Setiawan, M. (2023). Perbandingan Performa Apache Impala Dengan Apache Spark Dalam Mengeksekusi Kueri. *Journal of Network and Computer Applications*, *2*(2), 12–22. Retrieved from https://jurnal.netplg.com/

Atsila Imanda, R., Suroso, S., Fauzi, A., Simanjuntak, H. F., Azizah, Z., Destianty, A., … Zarka Zahira Shaffa, G. (2024). Pengaruh Data Warehouse Terhadap Pengambilan Keputusan. *Jurnal Portofolio : Jurnal Manajemen Dan Bisnis*, *3*(1), 31–39. Retrieved from https://www.jurnalprisanicendekia.com/index.php/portofolio/article/view/282

Bakari, R. I., Karamoy, H., & Lambey, R. (2022). Analisis Prosedur Pencairan Dana Langsung (LS) Pada Kantor Pelayanan Perbendaharaan Negara (KPPN) Manado. *Jurnal LPPM Bidang EkoSosKum(Ekonomi, Sosial, Budaya & Hukum)*, *5*(2), 941–948.

Darmansah, T., Agung, M. N., Hasbih, S. S., & Lucky Tirta, N. (2023). Tantangan dan Solusi dalam Pengelolaan arsip di era digital. *Jurnal Ekonomi Dan Bisnis Digital*, *02*(01), 5.

Eeti, S., GOEL, E. L., & KUSHWAHA, D. G. S. (2022). Efficient ETL Processes : A Comparative Study of Apache Airflow vs. Traditional Methods, *9*(8).

Fauzi, A., Noor, A. W., Ardyansyah, L. N., & Semesta, J. B. (2023). Kajian Penerapan Arsitektur Data Warehouse dalam Bisnis Intelijen pada Pengambilan Keputusan Bisnis. *JEMSI (Jurnal Ekonomi Manajemen Sistem Informasi)*, *4*(5), 868–875. Retrieved from https://www.dinastirev.org/JEMSI/article/download/1501/936

Irimia, C., Harbuzariu, F., Hazi, I., & Iftene, A. (2022). Official Document Identification and Data Extraction using Templates and OCR. *Procedia Computer Science*, *207*(Kes), 1571–1580. doi:10.1016/j.procs.2022.09.214

Jamaluddin, J., Nurfadila, N., & Isgunandar, I. (2023). Effectiveness of Archives Systems in Administrative Governance at the Maccini Sombala Village Head Office, Makassar City. *Pinisi Journal of Education and Management*, *2*(3), 265. doi:10.26858/pjoem.v2i3.56172

Kamdan, Somantri, Sundayana, M. G., & Kharisma, I. L. (2023). Rancang Bangun Layanan Private cloud Berbasis Infrastructure as a Service Menggunakan OpenStack dengan Metode Network Development Life Cycle(NDLC). *KLIK: Kajian Ilmiah Informatika Dan Komputer*, *4*(1), 252–262. doi:10.30865/klik.v4i1.1001

Li, C., Chen, Y., & Shang, Y. (2022). A review of industrial big data for decision making in intelligent manufacturing. *Engineering Science and Technology, an International Journal*, *29*, 101021. doi:10.1016/j.jestch.2021.06.001

Murtiwiyati, Hansel, A., & Leli, S. (2024). Implementasi Data Warehouse dan Business Intelligence Menggunakan Pentaho dan Metabase untuk Membuat Dahboard Visualisasi Kinerja Penjualan E-Commerce Wish. *Jurnal Penelitian Teknologi Informasi Dan Sains*, *Volume. 2*, 9.

Riza, N., Aulia, M. Z., Kolin, P. B., & Mustaqim, K. (2024). ANALISIS FAKTOR PENGARUH TERHADAP PENGHASILAN PROFESI DATA ENGINEER MENGGUNAKAN METODE REGRESI LINEAR BERGANDA. *JITET (Jurnal Informatika Dan Teknik Elektro Terapan)*, *13*(1), 9.

Sanchez, E. (2022). What Is Batch ETL Processing? The Only Guide You Need. Retrieved 23 May 2025, from https://blog.skyvia.com/batch-etl-processing/

Siti Sarah Sobariah Lestari, Gina Purnama Insany, Dede Sukmawan, & Faiz Dzulfikar Yusuf. (2023). Mengatasi Permasalahan High Dimensional Space dalam Klasifikasi Multikelas Big Data pada Data Gambar dengan DCSVM. *Jurnal RESTIKOM : Riset Teknik Informatika Dan Komputer*, *5*(3), 340–351. doi:10.52005/restikom.v5i3.259

Subekti, Z. M., Mukiman, K., Subandri, Sulthon, M. L., Sulistiyono, A., & Putra, R. E. (2024). RANCANG BANGUN INFRASTRUKTUR WEB SERVER. *JURNAL TRIDI*, *2*(1), 144–151.

Suriansyah, B., Mz, L. F., Rachman, A. I., & Pratiwi, G. (2025). Rekontruksi Arsitektur DataBase untuk Peningkatan Proses Load Data. *JURNAL MEDIA INFORMATIKA [JUMIN]*, *6*(2), 1455–1460.

Syuhada, H., Hidayat, S., Mulyati, S., & Giri Persada, A. (2023). Pengembangan Gamifikasi Pada Pelajaran Matematika Sd Dengan Metode Addie Untuk Meningkatkan Minat Belajar Siswa. *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, *9*(1), 1–14. doi:10.36341/rabit.v9i1.466

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. doi:10.20895/inista.v4i2.574

Wibawa, C., Wirawan, S., Mustikasari, M., & Anggraeni, D. T. (2022). Komparasi Kecepatan Hadoop Mapreduce Dan Apache Spark Dalam Mengolah Data Teks. *Jurnal Ilmiah Matrik*, *24*(1), 10–20. doi:10.33557/jurnalmatrik.v24i1.1649

Wulandari, F. A., & Maula, K. A. (2022). Analisis Sistem Akuntansi Penggajian Pada Badan Pusat Statistik (BPS) Kabupaten Bekasi PENDAHULUAN. *Jurnal Mirai Management*, *7*(2), 526–538. doi:10.37531/mirai.v7i2.2762

Yamjala, H. (2024). The Role of Data Engineering in AI and Machine Learning Projects. Retrieved 19 May 2025, from https://www.dataversity.net/the-role-of-data-engineering-in-ai-and-machine-learning-projects/

Yusuf Alfiansyah, F., & Arisandi, D. (2023). Perancangan Dashboard Monitoring Status Gizi Balita di Puskesmas Sukanagalih. *Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, 1–11.

Zulfadli, & Syahputra, R. (2024). SYSTEMATIC LITERATURE REVIEW: INTEGRATION OF BIG DATA AND ARTIFICIAL INTELLIGENCE. *JURNAL TEKNISI*, *4*(2), 40–47.