

Development of a Web-Based ETL Automation System to Support Monthly Reports

Ichha Maulidya^{1*}

^{1*}Software Engineering Technology, College of Vocational Studies, IPB University
ichamaulidya@apps.ipb.ac.id

Siti Farah Fakhirah², Jonser Steven Rajali Manik³, Asa Yuaziva⁴

^{2,3,4}Software Engineering Technology, College of Vocational Studies, IPB University

Abstract

The process of preparing monthly reports in the Asset Accounting Department of PT RS was hampered by manual data transformation and the lack of centralized storage, which caused historical data to be scattered, difficult to track, and vulnerable to human error. This condition prompted the development of a web-based Extract, Transform, Load automation system that integrates data cleansing, schema validation, a centralized repository, and a reporting dashboard as a replacement for external Business Intelligence tools. The methodology used was Rapid Application Development with Python (Flask) and MySQL. The system was tested through Alpha Testing with Black Box Testing using the Equivalence Partitioning technique for 34 test cases resulting in a validation percentage of 100%. Beta Testing to 8 respondents resulted in a feasibility of 96.4% (Admin) and 99.1% (User) in the "Very Feasible" category, as well as Performance Testing with an error rate of 0.00% at normal load and 0.30% at high load, as well as a stable throughput of 3.1 transactions per minute. These test results confirmed that the system met all acceptance criteria and was suitable to support a more efficient monthly report process.

Keywords: centralized repository, data cleansing, ETL, monthly report, rapid application development method, schema validation.

INTRODUCTION

The development of information technology in the current digital era has brought about significant changes in the way companies manage and utilize data (Maulidya et al., 2024; Mindara et al., 2025; Novianty et al., 2019; Nuryana et al., 2024; Siskandar et al., 2024; Siskandar, Mandang, et al., 2023a; Siskandar, Santosa, et al., 2022; Siskandar, Wiyoto, et al., 2022; Siskandar & Kusumah, 2019). Data no longer merely serves as transaction records but has become a strategic asset that plays a crucial role in supporting evidence-based business decision-making (data-driven decision-making) (Fatha et al., 2023; A. Hidayat et al., 2024; A. P. Hidayat et al., 2023; Hosen et al., 2024; Listy & Ilham, 2025; Santosa et al., 2021, 2023; Siskandar et al., 2023). In the era of digital transformation, every company is required to be able to integrate data from various sources quickly, accurately, and consistently so that the resulting information can be used to improve operational efficiency and reporting effectiveness (Basid et al., 2024; Sudiantini et al., 2023). To achieve this goal, a structured data management process is required through the Extract, Transform, Load (ETL) approach.

The ETL process plays a crucial role in extracting data from various sources, transforming it to suit analytical needs, and loading it into a centralized storage system such as a database or data warehouse (Firdaus & Shedriko, 2024; Prasetia & Kurniawan, 2021). The output of this ETL process subsequently becomes the foundation for Business Intelligence systems in generating reports and analyses of value to the company. One Business Intelligence platform commonly used by companies to build monthly-report dashboards is Microsoft Power BI, owing to its ability to turn raw data into interactive, informative visualizations (Fitriawan, 2024; Muku et al., 2023; Syaifullah, 2024). However, the quality of the resulting report depends heavily on the cleanliness, consistency, and structural conformity of the underlying data (Irzaman et al., 2022; Muzahra & Nasution, 2025;

Siskandar et al., 2020, 2021, 2023; Siskandar, Mandang, et al., 2023b). When input data is unclear, inconsistent with the established format, contains empty values, or fails to meet the required table structure, the reporting process can become inaccurate and prone to analytical and reporting errors (Muzahra & Nasution, 2025; Sartika & Aritonang, 2025).

Several prior studies have approached ETL automation and reporting from different angles, yet each solves only part of the problem. Putra et al., (2020) built an ETL engine with metadata and cleansing features for a bakery retailer, but the system was developed for a single company using the waterfall method and stopped short of a reporting dashboard or a centralized repository usable by non-technical staff. Putra et al., (2019) optimized ETL performance for government procurement reporting through a heuristic method, improving processing speed while leaving data cleansing, schema validation, and dashboard integration outside its scope. Putra & Putra, (2019) developed a Python-based ETL engine capable of automatic and manual execution, a technology choice close to Monica's, but it was designed as a general-purpose transformation tool rather than a platform with built-in validation rules and a reporting layer for end users. Nurfadhilah et al., (2025) designed a web-based dashboard to replace manual, paper-based production reporting, showing that web accessibility helps non-IT staff, yet the system was not backed by an automated cleansing or validation pipeline. Together, these studies indicate that ETL processing, data-quality assurance, and reporting visualization have largely been treated as separate concerns, with dashboards still depending on external Business Intelligence tools such as Power BI or Tableau for the final reporting layer. This points to a research gap: no existing study integrates data cleansing, schema validation, a centralized repository, and a reporting dashboard into a single web-based platform designed for users without an IT background, since existing ETL implementations still require a separate BI tool for visualization, while existing reporting dashboards are not supported by an automated data-quality pipeline.

This problem was also found at PT RS, a telecommunications company, specifically within its Asset Accounting Department. This department is responsible for regularly reporting asset and capitalization data every month. The process involves uploading data drawn from various raw files, such as Excel, into a database system. Before the data is uploaded, several adjustment steps are carried out, including column-format arrangement, removal of duplicate data, and verification of table-structure conformity. Because this process is performed manually, problems frequently arise as data volume increases, making the work slower, less efficient, and prone to input errors. The situation is further complicated by the fact that most staff do not have an information-technology background, and by the absence of a centralized data-storage system, which causes historical data to be scattered and difficult to trace. As a result, the monthly-report process often takes a long time and hinders timely data updates, a case that illustrates exactly the separation identified above, since Asset Accounting staff remain dependent on manual pre-processing before any report can be trusted.

In response to these problems, a web-based ETL automation system called Monica (Monitoring and Integration of Asset Capitalization) was developed. This system integrates data cleansing, schema validation, a centralized repository, and a reporting dashboard into a single, integrated platform to replace external Business Intelligence tools. A web-based system was chosen so that it could be easily accessed by non-IT staff and could support centralized data management. This emphasis on accessibility for non-technical users mirrors similar efforts elsewhere, such as the development of a smartphone-based smart communication system designed for residents without a technical background to improve neighborhood security response (Siskandar et al., 2025). The core features of automated data cleansing and schema validation ensure that the data-transformation process runs consistently, accurately, and in accordance with the structure required for monthly reporting, while data that has passed through the automated ETL process is stored centrally to simplify historical-data management, support audit needs, and ensure data quality. The system was developed using the Rapid Application Development (RAD) methodology with Python (Flask) and MySQL. This research aims to design and develop a web-based ETL automation system featuring data cleansing, schema validation, a MySQL-based centralized repository, and an integrated reporting

dashboard to improve the speed, accuracy, and standardization of the monthly reporting process in the Asset Accounting Department of PT RS. The novelty of this research lies in integrating automated data cleansing, schema validation, a centralized repository, and a reporting dashboard within a single web-based platform for users without an IT background, an integration not addressed jointly by the ETL and reporting-automation studies discussed above.

METHODS

This research, was conducted over seven months, from October 2025 to April 2026, in two locations: the Asset Accounting Department of PT RS as the data-collection environment, and the Electromechanical Workshop Laboratory, College of Vocational Studies, IPB University, as the system-development environment. Data were collected using three approaches: a literature study, a focus group discussion, and participatory observation.

The literature study was carried out continuously throughout the research period by reviewing relevant national and international scientific journals, textbooks, and prior undergraduate final reports, focusing on five main topics that form the theoretical foundation of this research: the concept and implementation of the ETL process, data-cleansing and schema-validation mechanisms, the Rapid Application Development methodology, web-based system-development technology using Python and Flask, and system evaluation through Black Box Testing, User Acceptance Testing, and Performance.

Focus group discussions were conducted in a hybrid format at PT RS, involving the General Manager, Managers, Supervisors, staff, and Admin of the Asset Accounting Department, who were selected purposively because they are directly involved in the monthly report preparation process and in managing asset capitalization data. The discussions followed a structured guide that focused on four main topics: the current data management workflow, obstacles in the manual data transformation process, business rules applied in asset data processing, and user expectations for the proposed automated system. The discussion results were analyzed and formulated as user stories following the standard format "As [user role], I want [required features], so that [expected benefits]," which served as the basis for identifying the system's functional requirements in the Requirements Planning phase. The resulting user stories are shown in Table 1.

Table 1. User stories

Code	Actor	User story
US01	Admin	As an Admin, I want to log in to the system so that I can access data-management features according to my role.
US02	User	As a User, I want to log in to the system so that I can access reporting dashboards according to my role.
US03	Admin	As an Admin, I want to flexibly upload Excel files through the web interface so that I do not need to upload directly to a complex server.
US04	Admin	As an Admin, I want the system to let me select the active sheet and target database table before upload, so that incoming data reaches the correct table.
US05	Admin	As an Admin, I want the system to automatically validate column structure, data type, and column order, so that input data always matches the target table schema.
US06	Admin	As an Admin, I want the system to perform automatic data cleansing (deduplication, whitespace, dates, missing values, reference mapping), so that I no longer need manual transformation in Excel.
US07	Admin	As an Admin, I want a preview of validation and cleansing results so that I can confirm data correctness before saving.
US08	Admin	As an Admin, I want to monitor the history and status of every upload (success/failure), so that I can trace and follow up on failures.

US09	Admin	As an Admin, I want to view and manage asset data (GR, GR0, Unpaid/FPJP, PO Absorption, Settlement, ZPS AUC) in one platform, so that data is easy to access and analyze.
US10	Admin	As an Admin, I want to download historical data from the database in Excel format, so that I can use it for further analysis outside the system.
US11	Admin	As an Admin, I want to manage reference data (Master COA, Master GL Account, Capex Paper, Capex Master) in one platform, so that master data can serve as a consistent ETL reference.
US12	Admin	As an Admin, I want to manage user accounts and roles, so that I can determine each user's access rights.
US13	Admin	As an Admin, I want to view the Asset Capitalization Report dashboard with a month/year period filter, so that I can visually monitor assets and capitalization in one view.
US14	User	As a User, I want to view the Asset Capitalization Report dashboard with a month/year period filter, so that I can visually monitor the status of assets and capitalization.
US15	Admin	As an Admin, I want to view the Monitoring Capex dashboard with several report tabs, so that I can regularly monitor capex progress and status in one view.
US16	User	As a User, I want to view the Monitoring Capex dashboard with several report tabs, so that I can regularly monitor capex progress and status.
US17	Admin	As an Admin, I want to download dashboard reports in PPTX and Excel format, so that I can use them for presentation and further analysis.
US18	User	As a User, I want to download dashboard reports in PPTX and Excel format, so that I can use them for historical purposes.
US19	Admin	As an Admin, I want to log out of the system, so that I can safely end my session.
US20	User	As a User, I want to log out of the system, so that I can safely end my session.

Participatory observation was carried out directly in the Asset Accounting Department of PT RS during an internship period from October to December 2025, through four activities: assisting staff with data cleansing and column-format adjustment, analyzing the original working files used in the monthly-report process, participating in the team's daily meetings and discussions, and holding direct question-and-answer sessions with staff regarding the specific business rules used in mapping data between tables. The observation showed that data preparation including cleansing, format standardization, and VLOOKUP-based mapping, was still performed manually in Microsoft Excel, even though data storage and visualization already relied on a database and Power BI. Dependence on VLOOKUP as an inter-table mapping mechanism proved technically limiting, since the formula is static with respect to the position of the reference column, so staff had to rebuild it manually whenever the column structure changed; the absence of automated format-conformity validation also caused errors, such as inconsistent date formats, to be discovered only after the final report had been compiled. The database-upload process itself was also performed manually through server access requiring technical understanding of directories and database mechanisms, creating an operational dependency on specific staff. These findings formed the basis for designing the web-based ETL automation system and for defining the business rules, data-cleansing mechanisms, and schema-validation rules applied to each data table.

Based on this analysis, Rapid Application Development (RAD) was selected as the appropriate system-development method, because its iterative approach allows gradual adjustment to changing requirements while encouraging active user involvement at every development stage (Agarwal et al., 2017; Stephens, 2012), so that the resulting system could better match the business processes running in the Asset Accounting Department of PT RS. Similar rapid, iterative development

approaches have also proven effective in other web-based service platforms developed within the same institution, such as a digital reservation and scheduling system built with the Prototyping method for museum visitor management, which likewise replaced a manual, paper-based process with a structured web application (Manik et al., 2026). This research applied two main methods, system design and testing, to ensure the system met the functional requirements and operated effectively in its operational environment. The system-design method followed the phases of RAD, from Requirements Planning and User Design to Construction, while the testing method was carried out as part of the cutover phase of RAD, the final validation stage prior to hand-over to users, encompassing both technical functional verification and measurable user-acceptance evaluation.

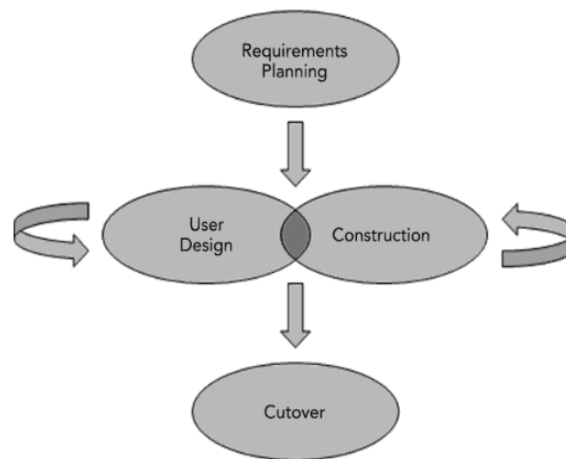


Figure 1. Rapid Application Development stages

System Design

System design was carried out according to the three phases of the Rapid Application Development methodology, namely Requirements Planning, User Design, and Construction. Each phase formed the foundation for the next, creating a structured, adaptive, and verifiable development cycle.

a) Requirements Planning

In the Requirements Planning phase, information gathered from literature studies, focus group discussions, and participant observation is analyzed to produce a system functional requirements specification in which each functional requirement code is mapped to its corresponding user story code, complete with its function name and description. This covers the system's core features ranging from data upload and transformation to master data management and integrated reporting dashboards, as shown in Table 2.

Table 2 Functional requirements (KF) mapped to user-story codes

US Code	KF code	Function	Description
US01– US02	KF01	Login	Provides user authentication using a username and password to access the system according to the user's role.
US03	KF02	Upload data file	Allows the Admin to upload an Excel file (XLSX/XLS/XLSM) through the web interface via drag-and-drop or browse.
US04	KF03	Select sheet and target table	Allows the Admin to select the sheet and target database table before the upload process begins.

US05	KF04	Schema validation	Automatically validates column structure, data type, and column order of the uploaded file against the target table schema.
US06	KF05	Automated data cleansing	Automatically cleans data, including duplicate removal, whitespace trimming, date-format standardization, missing-value handling, and column mapping.
US07	KF06	View validation and cleansing results	Displays the schema-validation and data-cleansing report to the Admin before data is loaded into the database.
US08	KF07	Upload history	Stores and displays the history of every upload process together with its status (success/failure).
US09	KF08	Asset data management	Allows the Admin to view and manage historical asset data in one platform.
US10	KF09	Download data to Excel	Allows the Admin to download data stored in the database in Excel format for further analysis.
US11	KF10	Master-data management	Allows the Admin to manage the reference master data used in the ETL process.
US12	KF11	Account management	Allows the Admin to manage user accounts and roles, including adding, editing, and deleting accounts.
US13– US14	KF12	Asset Capitalization Report dashboard	Displays the Asset Capitalization Report dashboard, accessible by Admin and User, with a month/year period filter and six report tabs.
US15– US16	KF13	Monitoring Capex dashboard	Displays the Monitoring Capex dashboard, accessible by Admin and User, with five report tabs.
US17– US18	KF14	Download report	Allows dashboard reports to be downloaded in PPTX and Excel format.
US19– US20	KF15	Logout	Allows the user to log out of the system to end the session and maintain account security.

Beyond these functional requirements, two non-functional requirements were also defined based on conditions found during observation and the focus group discussion: usability, requiring the system to be easy enough for staff to upload data effectively and in accordance with business rules, and data consistency, requiring processed data to remain consistent with the established structure so that it is immediately usable for reporting without additional adjustment. These requirements were implemented on a development stack consisting of Python, HTML, CSS, and JavaScript as the programming languages, Flask as the backend framework, and MySQL as the database, running on Windows 11.

b) User design

The User Design phase translates these user stories into a system architecture consisting of use case diagrams, activity diagrams, sequence diagrams, and entity-relationship diagrams, along with user interface designs, all of which are iteratively validated with stakeholders. The use case diagram is shown in Figure 2.

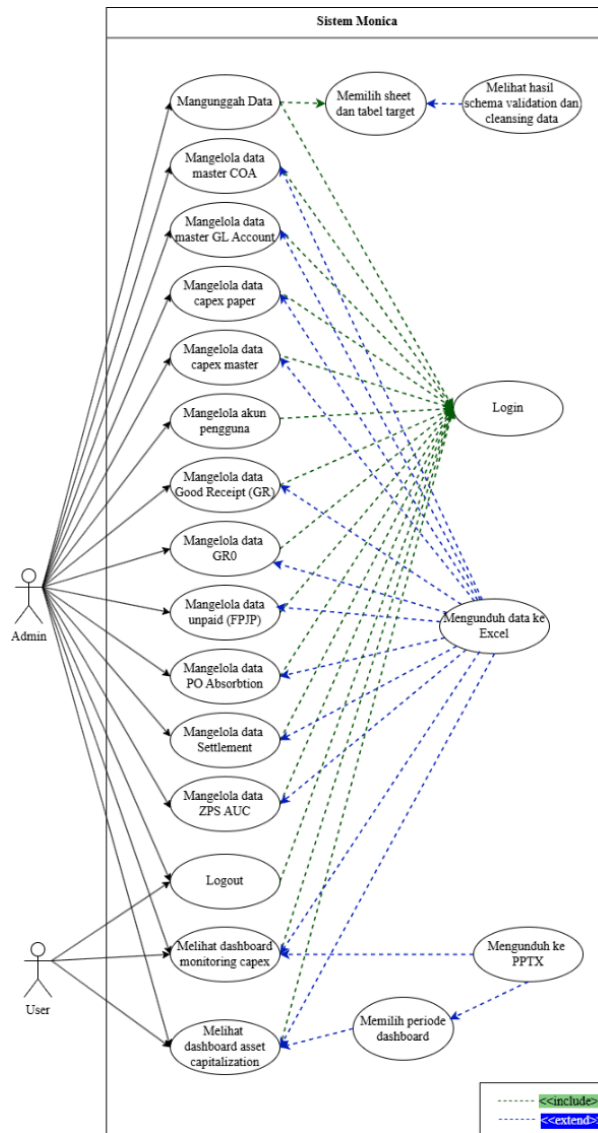


Figure 2 .Use case diagram

Figure 2 illustrates the interaction between the system and its two primary actors: the Admin and the User. The Admin represents the Asset Accounting Department team responsible for overall system management, while the User represents stakeholders such as the General Manager, Vice Presidents, and other departments that require access to reporting information. Both actors must log in before accessing system features through an <<include>> relationship. The Admin has full access to data upload, master data, asset data, user management, and dashboards. During data upload, the system automatically performs schema validation and data cleansing after the target sheet and table are selected, with the validation results available through an <<extend>> relationship. The Admin can also export managed data in Excel format. In contrast, the User can only access the Capex Monitoring Dashboard and the Asset Capitalization Dashboard, which supports period filtering and report export in PPTX and Excel formats through an <<extend>> relationship. To support these system functions and the underlying ETL process, the database structure was subsequently designed using an entity-relationship diagram, as shown in Figure 3.

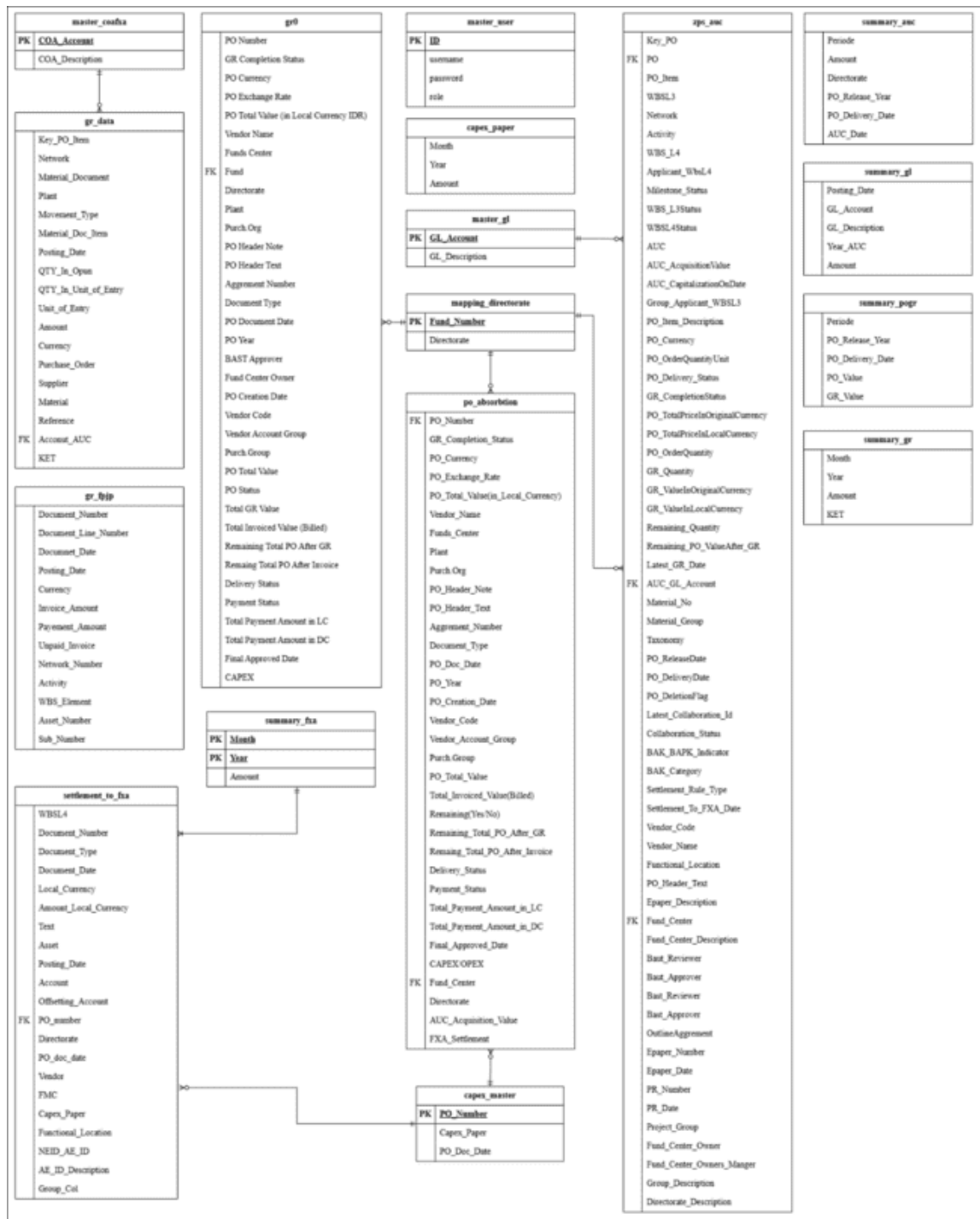


Figure 3. Entity-relationship diagram of the Monica database

The database schema was designed using an entity-relationship diagram (Figure 3) to support the ETL process, dashboard reporting, and user-access management within a single integrated database. The schema consists of transactional asset-capitalization tables, master/reference tables, a user table, and summary tables. These tables are connected through keys such as **PO_Number**, **COA_Account**, **GL_Account**, and **Fund_Number** to maintain data consistency during ETL processing. For example, **gr_data** references **master_coafx** through the **Account_AUC**–**COA_Account** relationship, while **gr0** and **po_absorption** reference **mapping_directorate** through the **Fund/Fund_Center** fields to populate directorate information automatically. The **master_user**, **capex_paper**, and **summary** tables remain independent because they either support standalone functions or store the final outputs of the ETL process. Based on the functional and database designs, the user interface was then modeled using low-fidelity wireframes.

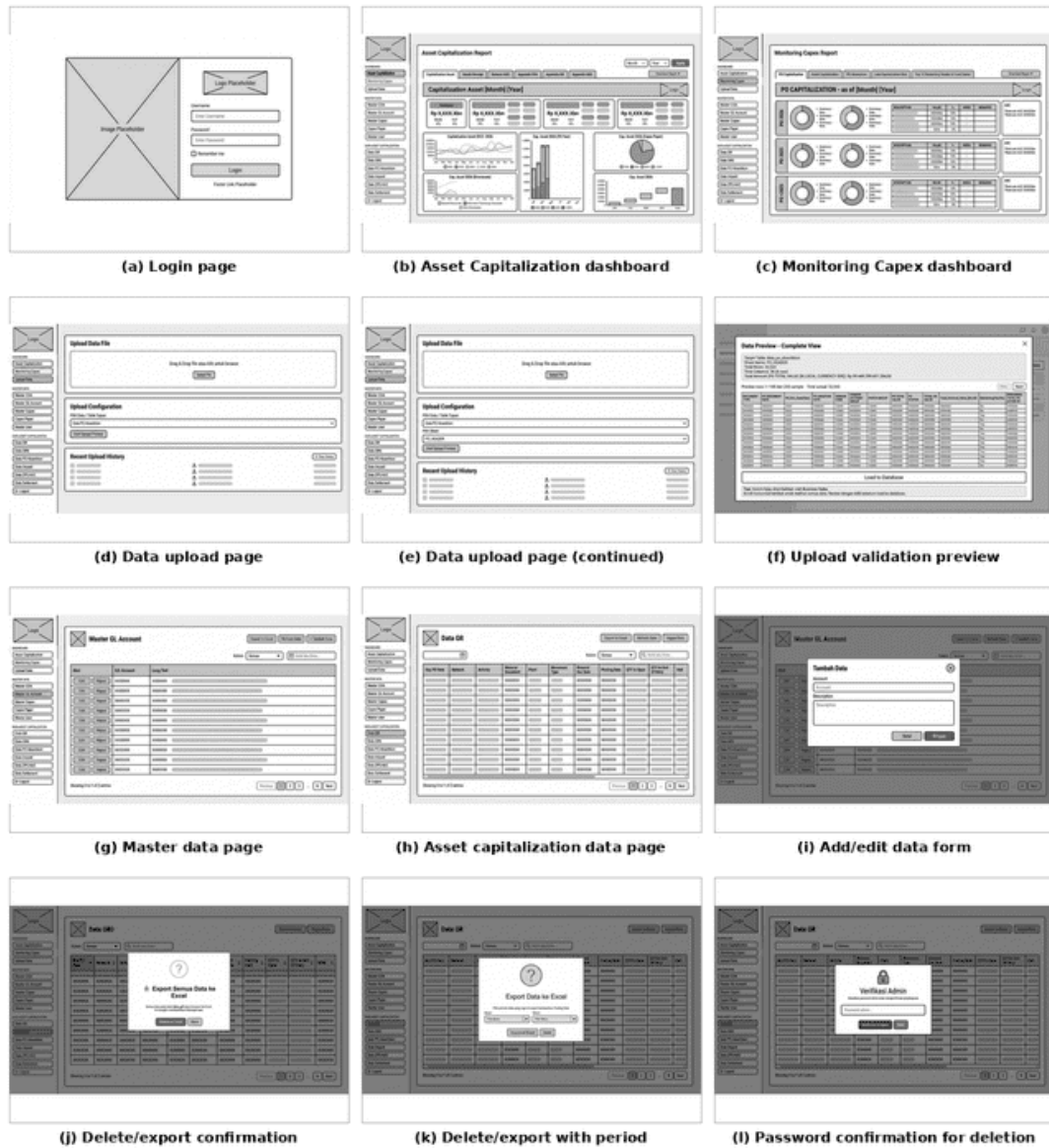


Figure 4. Low-fidelity wireframes

Then, a low-resolution framework was designed for each of the system's main pages, including the login page, data upload, master data, asset capitalization data, two dashboards, and system notifications, focusing on the placement of key elements such as navigation, content areas, data tables, charts, and interactive components without considering color, typography, or other visual details as shown in Figure 4. This framework was reviewed and approved by all stakeholders, so they could validate the navigation flow and completeness of information before the interface proceeded to more detailed visual development.

c) Construction

The Construction phase implemented the system components defined during the User Design phase, including user authentication, the automated ETL pipeline, master-data management, asset-capitalization data management, and reporting dashboards. The primary focus was the implementation of the ETL pipeline, which performs data extraction, schema validation, business-rule-based transformation, data cleansing, loading into the centralized database, and automatic summary generation. The pipeline was designed to process SAP export files used by the Asset Accounting Department of PT RS, whose column order, column names, date formats, and representation of empty values may differ across reporting periods. The workflow begins when the

Admin uploads an Excel file, selects the target table and, if necessary, the worksheet to be processed. The system then executes the Extract, Transform, and Load stages sequentially.

1) Extract

During the Extract stage, the system first identifies the available worksheets and automatically detects the header row instead of assuming that the first row contains column names. This approach accommodates SAP export files that frequently contain title rows, blank rows, or additional information before the actual dataset. The header is selected by evaluating the first 15 rows and choosing the row with the highest number of non-empty textual cells. After the header is established, empty rows are removed and the system generates an initial preview containing the total number of rows, the total value of the Amount column when available, and a sample of 200 records. The extracted dataset is cached temporarily on the server to avoid repeated uploads during subsequent processing. Before entering the Transform stage, the frontend validates the uploaded columns against the required schema for the selected target table and warns the Admin when more than 20% of the required columns are missing.

2) Transform

The Transform stage applies business rules obtained through observation and focus group discussions to convert the extracted data into a standardized format suitable for storage. The same transformation logic is executed twice, first on the frontend to generate a preview and again on the backend during the final loading process. Generic transformations include column-name mapping, numeric and date standardization, normalization of empty values, duplicate removal, whitespace trimming, and missing-value handling. Additional rules are applied according to the data category, including automatic directorate mapping based on Fund or Fund Center values and validation of COA and GL account references. The transformation results are displayed in a preview showing the target table, worksheet, row and column counts, total amount, and system-generated columns, allowing the Admin to verify the processed data before it is stored.

3) Load

After the Admin confirms the preview, the transformed data is stored in the centralized MySQL repository using either Replace or Append mode according to the data category. The system then updates the corresponding summary tables to support efficient dashboard reporting without repeatedly querying the transactional tables. Every upload attempt, whether successful or unsuccessful, is recorded in an upload history log to provide complete traceability of data processing activities. Through this integrated ETL workflow, Monica replaces manual spreadsheet processing with a standardized, browser-based process that improves consistency, reduces manual effort, and supports faster monthly reporting.

Testing Method

System testing was performed as part of the RAD cutover phase, the final validation stage prior to hand-over to users, and consisted of Alpha Testing to verify the system's technical functionality, Beta Testing to measure user acceptance, and Performance Testing to evaluate the system's resilience under concurrent user load.

a) Alpha Testing

Alpha Testing was conducted as an internal validation to evaluate the system's technical performance using Black Box Testing with the Equivalence Partitioning technique, which is well suited to a web-based ETL system that processes a wide range of valid and invalid inputs. The testing covered six main features, namely Login, Data Upload, Dashboard, Account Management, Master Data, and Asset Capitalization Data. The system was considered successful when every test scenario produced the expected output, following the validation-percentage formula proposed by (Hardika et al., 2024).

$$Persentase Validasi = \left(\frac{Jumlah\ Test\ Case\ Valid}{Total\ Test\ Case} \right) \times 100\%$$

b) Beta Testing

Beta Testing was an external validation representing the perspective of Asset Accounting end-users, conducted through a Likert-scale questionnaire covering three aspects: user acceptance, usability, and perceived time-efficiency relative to the manual process (Hambling & Goethem, 2013). The questionnaire was administered to a total of eight respondents, divided into two groups according to their role: five Admin respondents from the Asset Accounting Department, who completed a questionnaire covering eight evaluation categories, and three User respondents from outside the department, who completed a shorter questionnaire covering four evaluation categories relevant to their role as report and dashboard consumers. Each statement was measured using a five-point Likert scale, where a score of 1 represented Strongly Disagree and a score of 5 represented Strongly Agree, as presented in Table 3.

Table 3. Likert scale score weighting scale

Scale	Answer	Score
STS	Strongly Disagree	1
TS	Disagree	2
N	Neutral	3
S	Agree	4
SS	Strongly Agree	5

The questionnaire results were first processed by calculating the mean score for each statement and evaluation aspect. The mean score was calculated using the following formula.

$$\text{Mean score} = \frac{\text{Total Likert Score}}{\text{Number of Respondents}}$$

Based on the calculated mean values, the percentage score was determined to measure the overall acceptance level of the system using the following formula.

$$\text{Percentage (\%)} = \left(\frac{\text{Mean Obtained}}{\text{Maximum Likert Score}} \right) \times 100\%$$

Based on this calculation, the resulting percentage was classification into five levels, namely Very Infeasible, Infeasible, Fairly Feasible, Feasible, Very Feasible, as shown in Table 4. These classification were used to determine the overall acceptance level of the developed Monica system.

Table 4. Feasibility Percentage Interval Classification

Confidence Interval (%)	Category
0-19,99	Very Infeasible
20-39,99	Infeasible
40-59,99	Fairly Feasible
60-79,99	Feasible
80-100	Very Feasible

The UAT questionnaire was designed to evaluate the system from three main aspects: user acceptance, usability, and time efficiency compared with the previous manual process. These aspects were represented through eight evaluation categories covering authentication and access security, ETL process and data cleansing, centralized storage and data traceability, asset-capitalization data management, reporting dashboards, suitability of system needs and pain points, and user interface usability. The detailed evaluation categories and the number of test cases are presented in Table 5.

Table 5. UAT Testing Categories

No	Category	Number of Test Case
1	Authentication and Access Security (Login)	3
2	ETL Process and Data Cleansing	9
3	Centralized Storage and Data Traceability	6

4	Data Management (Master Data Asset Capitalization)	7
5	Asset Capitalization Report Dashboard	6
6	Capex Monitoring Dashboard	6
7	Suitability of Needs and Pain Points	7
8	Appearance and Ease of Use (UX)	5

The statements in both questionnaires were developed by the researcher based on the system requirements and user stories defined during the Analysis phase (Table 1) and were grouped into the evaluation categories shown in Table 5, following the general user-acceptance framework of (Hambling & Goethem, 2013). This approach is consistent with prior UAT-based feasibility studies of financial and administrative information systems, which similarly use Likert-scale questionnaires interpreted through percentage-based feasibility categories (Aliyah et al., 2024). Item wording was reviewed together with the thesis supervisor prior to distribution to check clarity and relevance to each category. Formal statistical validation of the instrument, such as content-validity judgment or reliability testing (e.g., Cronbach's alpha), was not performed, since the questionnaire was intended as a practical user-acceptance evaluation rather than a psychometrically validated research instrument; this is acknowledged as a limitation of the present study. Given this limitation and the small number of respondents in each group, the resulting feasibility percentages, including the "Very Feasible" classification reported in the following section, should be read as an indication of user acceptance under the tested conditions rather than as statistically generalizable evidence of the system's effectiveness.

The evaluation involved respondents selected using purposive sampling, where participants were intentionally chosen based on their relevance, knowledge, and direct involvement with the tested system (Kusuma & Yufon, 2024). This approach follows the UAT principle proposed by (Hambling & Goethem, 2013), which states that testing should involve the intended users of the system. Therefore, respondents were selected from the Asset Accounting Department based on their direct interaction with Monica in operational activities and reporting processes.

The UAT evaluation involved eight respondents divided into two user groups. The first group consisted of five Admin users responsible for system operations, including ETL processing, master-data management, and asset-capitalization data management. The second group consisted of three User representatives, including one General Manager and two representatives from external divisions responsible for monitoring dashboards and periodic asset-accounting reports.

The final UAT results were calculated by averaging the scores obtained from all statements and respondents. The resulting percentages were then interpreted using predetermined feasibility percentage intervals to determine the system acceptance level. The system was considered accepted when the obtained percentage reached the minimum acceptance threshold of 61%, which corresponds to the Feasible category according to (Hakky et al., 2018).

c) Performance Testing

Performance Testing evaluated the reliability, response time, and processing consistency of the ETL pipeline API endpoints using Apache JMeter. Testing was performed at the backend layer because processing speed and stability at this layer directly determine the user experience during data upload and dashboard rendering. Two testing scenarios were conducted. The first was a load-testing scenario simulating five concurrent Admin users with a timeout of 600,000 ms (10 minutes). The second was a stress-testing scenario simulating fifteen concurrent users, equivalent to three times the normal load, without a timeout limit to evaluate the maximum resilience of the system architecture.

RESULTS AND DISCUSSION

System Implementation

Based on the system design described above, Monica (Monitoring and Integration of Asset Capitalization) was implemented as a web-based application built with Python (Flask) as the backend, MySQL as the centralized database, and HTML, CSS, and JavaScript for the user interface, accessible

through a browser without any additional software installation. Figure 5 presents the resulting interface for the system's main pages, from the authentication gate through to the reporting dashboards.

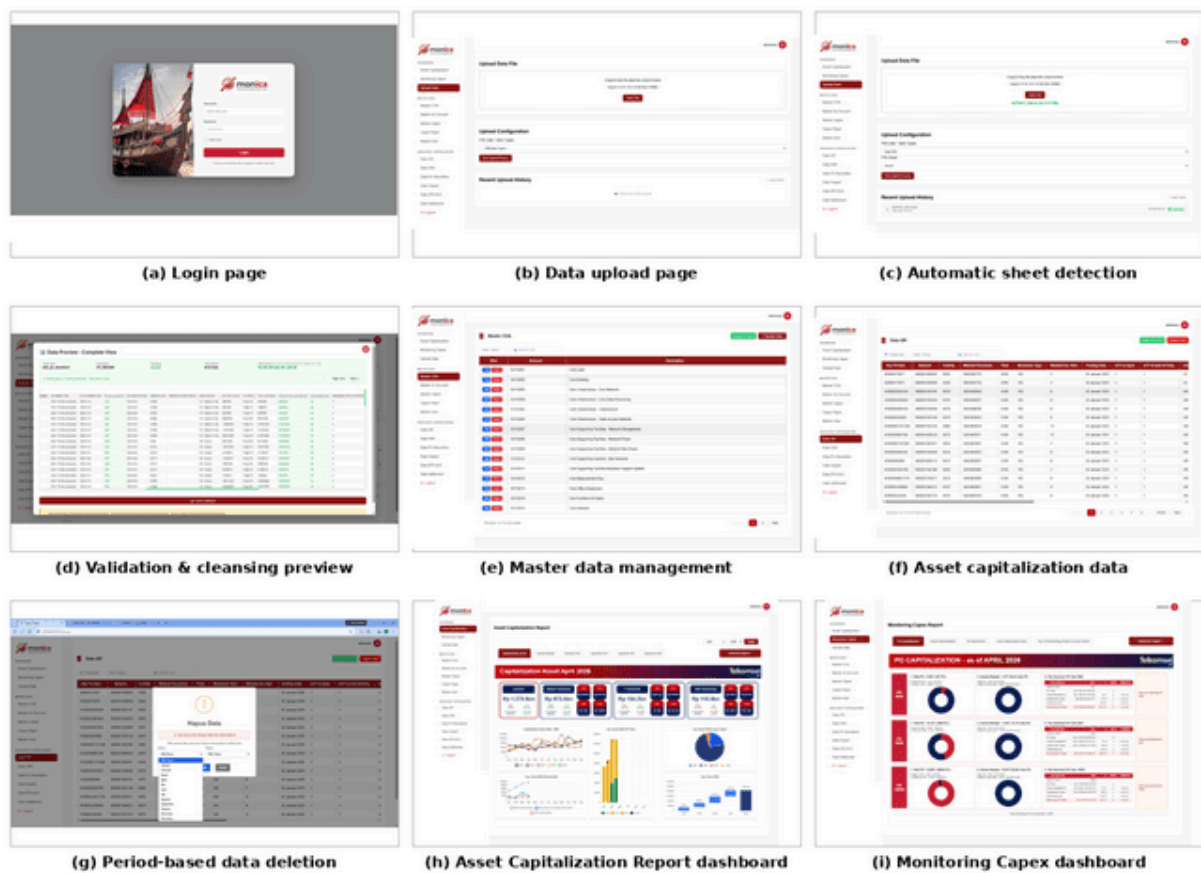


Figure 5. Monica web interface: (a) login, (b–c) data upload and automatic sheet detection, (d) validation and cleansing preview, (e) master data, (f) asset-capitalization data, (g) period-based data deletion, (h–i) reporting dashboards

The login page shown in Figure 5a serves as the authentication gateway for both Admin and User accounts. After successful authentication, each role is directed to its corresponding dashboard according to its access permissions. The data upload interface in Figure 5b and Figure 5c, together with the validation and cleansing preview in Figure 5d, provides the frontend implementation of the ETL pipeline described in the Construction phase. Through this interface, the Admin selects the worksheet and target table before reviewing the results of schema validation and data cleansing prior to loading the data into the centralized repository.

Master data management is provided through a dedicated interface shown in Figure 5e, covering Master COA, Master GL Account, Capex Paper, Capex Master, and user accounts. The page supports searching, filtering, adding, editing, deleting with password verification, and exporting data to Excel. Asset-capitalization data management, illustrated in Figure 5f, provides similar management functions for GR, GR0, Unpaid, PO Absorption, Settlement to FXA, and ZPS AUC data. Deletion can be performed either by reporting period or directly on selected records after password verification, as shown in Figure 5g, and the processed data can also be exported to Excel.

Reporting is provided through two integrated dashboards that use the summary tables generated during the Load stage. The Asset Capitalization Report dashboard, presented in Figure 5h, supports month and year filtering and consists of six report tabs, namely Capitalization Asset, Good Receipt, AUC Balance, Appendix Capitalization Asset, Appendix GR, and Appendix AUC. The Monitoring Capex Report dashboard, shown in Figure 5i, consists of five report tabs including PO Capitalization, Asset Capitalization, PO Absorption, Late Risk Cap, and Top 10 Remaining. Unlike

the previous reporting process that relied on Power BI, both dashboards are updated automatically after every successful ETL process and allow reports to be exported in PPTX and Excel formats for presentation and further analysis.

Testing Results

System testing was carried out as the final validation stage prior to hand-over to users, consisting of Alpha Testing using Black Box Testing, Beta Testing using User Acceptance Testing, and Performance Testing using Apache Jmeter

a) Alpha Testing Results

Alpha Testing was conducted on 34 test cases distributed across the six main features, as presented in Table 6. All test scenarios, including valid and invalid login credentials, single- and multi-sheet Excel uploads, unsupported file formats, missing-value handling, duplicate handling, date-format standardization, dashboard period filtering, report downloads, and CRUD operations secured by password verification for master data, user accounts, and asset-capitalization data, produced outputs that matched the expected results. The testing achieved an overall validation percentage of 100%, confirming that the authentication module, ETL pipeline, master-data management, and dashboard functionalities of Monica operate according to the specified functional requirements. Therefore, the system was considered ready to proceed to Beta Testing.

Table 6. Recapitulation of Black Box Testing results

No	Feature	Test cases	Valid (%)
1	Login	5	100
2	Data upload (ETL pipeline)	7	100
3	Dashboard	4	100
4	Master data	7	100
5	Account management	4	100
6	Asset capitalization data	7	100

b) Beta Testing Results

Beta Testing (UAT) was conducted with eight respondents consisting of five Admin staff from the Asset Accounting Department of PT RS and three User respondents from outside the department who interact with Monica as dashboard and report consumers. The evaluation results from the Admin respondents are presented in Table 7.

Table 7. Recapitulation of Beta Testing (UAT) results for Admin

No	Category	Items	Mean	Feasibility (%)
1	Authentication and access security	3	4,93	98,6
2	ETL process and data cleansing	9	4,88	97,6
3	Centralized storage and data traceability	6	4,60	92,0
4	Data management (master data)	7	4,83	96,6
5	Asset Capitalization Report dashboard	6	4,83	96,6
6	Monitoring Capex dashboard	6	4,83	96,6
7	Requirement fit and pain points	7	4,86	97,2
8	Display and ease of use (UX)	5	4,80	96,0

Based on the results shown in Table 7, the Admin respondents gave Monica an overall mean score of 4,82 out of a maximum Likert score of 5. This score corresponds to a feasibility percentage of 96,4%, which falls within the 80–100% interval and is categorized as Very Feasible. The highest

evaluation scores were obtained for Authentication and Access Security with 98,6% and the ETL and Data-Cleansing process with 97,6%, followed by Requirement Fit and Pain Points with 97,2% and Data Management and Dashboards with 96,6%. The results indicate that Monica successfully automated Excel data extraction, duplicate detection, column-format standardization, and schema-validation notifications. Furthermore, the Admin respondents confirmed that the system provided centralized data management and reduced dependency on scattered local files during the reporting process.

In addition to the Admin evaluation, Beta Testing also involved User respondents who assessed Monica from the perspective of report consumers and dashboard users. The evaluation results from the User respondents are presented in Table 8.

Table 8. Recapitulation of Beta Testing (UAT) results for User

No	Category	Items	Mean	Feasibility (%)
1	Authentication and access security	3	5,00	100
2	Asset Capitalization Report dashboard	6	4,88	97,6
3	Monitoring Capex dashboard	6	4,94	98,8
4	Display and ease of use (UX)	5	5,00	100

Based on the results shown in Table 8, the User questionnaire produced an overall mean score of 4,95 out of 5, corresponding to a feasibility percentage of 99,1%, which falls within the Very Feasible category. The highest-rated aspects were Authentication and Access Security and Display and Ease of Use, both achieving 100%, followed by the Monitoring Capex Dashboard with 98,8% and the Asset Capitalization Report Dashboard with 97,6%. These results indicate that Monica can be accessed and utilized by users outside the Asset Accounting Department without requiring specialized technical training.

The Beta Testing results from both respondent groups demonstrate that the three research objectives were achieved. The first objective, developing a web-based ETL automation system with data-cleansing and schema-validation features, was supported by respondents' confirmation that Monica simplified data preparation and reduced potential human errors compared with the previous manual process. The second objective, implementing a centralized repository to maintain historical data consistency, was supported by the ability to access previous-period data through a single integrated storage system without relying on separate local files. The third objective, providing an integrated web-based reporting dashboard as an alternative to external Business Intelligence tools, was supported by the evaluation results of the Asset Capitalization Report and Monitoring Capex dashboards from both respondent groups. These results indicate that Monica successfully provides centralized reporting capabilities within a single platform. With feasibility percentages exceeding the 61% minimum acceptance threshold, Monica was considered accepted by users and suitable for supporting the operational reporting process of the Asset Accounting Department of PT RS.

Regarding the objective of improving reporting efficiency, a direct quantitative comparison was not part of the experimental design; the comparison below is therefore descriptive, drawn from the participatory observation described in the Analysis phase, rather than from a controlled before-and-after measurement. As reported by Asset Accounting staff, the previous manual process, comprising SAP data extraction, Excel-based consolidation using VLOOKUP formulas, and manual compilation into a reporting dashboard, typically took approximately three to five working days per monthly cycle. In comparison, Alpha and Performance Testing showed that Monica's automated ETL pipeline, which replaces the manual extraction and consolidation steps, completed data validation, cleansing, and loading within minutes under normal operating load. This gap in processing time is consistent with respondents' feedback in the Beta Testing evaluation and suggests a substantial reduction in manual effort, although it should be interpreted as an indicative comparison rather than a

rigorously controlled experimental result, since the manual-process duration was based on staff recollection rather than instrumented measurement.

c) Performance Testing Results

The first scenario simulated daily usage by five Admin users with a 600.000 ms (10-minute) timeout. Based on the Apache JMeter summary report shown in Figure 6, a cumulative total of 225 requests were recorded, and every functional component of Monica achieved a 0,00% error rate. This result indicates that under normal load conditions, the system processed all ETL transactions without connection failures or HTTP errors, meeting the ideal 0% error-rate criterion. In terms of response time, the highest average value was recorded by the Parse Data FXA module at 438.147 ms (7,3 minutes). Although this process required several minutes due to the large volume of Excel data being processed, it was completed within the defined 10-minute timeout limit. The system also maintained a stable throughput of 3,1 transactions per minute, indicating adequate performance to support the Admin's daily operational activities.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	15	1.144	308	5210	1408.94	0.00%	17.5/sec	0.05	0.00	9807.7
Parse Data GRD	15	3.776	1424	8796	2472.15	0.00%	17.5/sec	0.21	0.10	40389.3
Mapping Discriminator ...	15	10.983	1291	23946	8810.83	0.00%	17.5/sec	0.21	0.00	40389.3
Upload Data GRD	15	17.018	4289	24785	8321.49	0.00%	17.5/sec	0.06	0.00	760.8
Parse Data GR	15	26877.0	210444	351423	54587.37	0.00%	16.5/sec	0.30	41.11	47702.8
Mapping Discriminator ...	15	4.388	270	8725	1577.86	0.00%	16.4/sec	0.22	0.00	40389.3
Upload Data GR	15	32737	28628	84533	4388.08	0.00%	16.2/sec	0.00	0.00	5775.0
Parse Data FXA	15	438.147	385476	477680	38687.79	0.00%	16.0/sec	0.33	81.15	75440.1
Mapping Discriminator ...	15	41.80	462	8546	1988.58	0.00%	16.0/sec	0.21	0.00	40389.3
Upload Data FXA	15	195880	64423	311339	87543.08	0.00%	16.5/sec	0.00	0.00	437.8
Parse Data PG Abc	15	146111	60750	338097	27121.39	0.00%	16.4/sec	0.33	33.72	138115.7
Mapping Discriminator ...	15	36.85	78	9627	1586.48	0.00%	17.2/sec	0.20	0.00	40389.3
Upload Data PG Abc	15	27880	13254	12446	5203.48	0.00%	17.1/sec	0.00	0.00	386.8
Parse Data DPS AGC	15	193312	116485	234512	37591.29	0.00%	16.4/sec	0.17	47.11	162468.7
Upload Data DPS AGC	15	81983	72836	181811	6786.77	0.00%	17.1/sec	0.00	0.00	5753.0
TOTAL	225	94884	270	477680	129102.00	0.00%	3.1/min	2.28	153.22	458149.8

Figure 6. Performance testing result with 5 concurrent users

The second scenario tested the maximum resilience of the system architecture by increasing the load to fifteen users, which represents three times the normal capacity, with no timeout limit. Based on the Apache JMeter results shown in Figure 7, the stress-testing scenario generated 675 simultaneous requests. Under this extreme-load condition, Monica maintained a high level of reliability with an overall error rate of only 0,30%. This result remained close to the ideal 0% error-rate criterion, indicating that the server architecture remained stable despite a significant increase in request volume. The response time of the Parse Data FXA module increased to 1.440.897 ms (24 minutes), compared with 438.147 ms (7,3 minutes) under normal load. This increase occurred due to resource contention, where fifteen simultaneous virtual users competed for the available CPU and memory resources. The impact was more significant on this module because its process is CPU-intensive and involves reading and processing large-volume Excel files. Nevertheless, the process completed without any global connection failure, demonstrating that the server architecture maintained operational stability under high-load conditions.

Summary Report												
Name:	Summary Report											
Comments:												
Write results to file / Read from file												
Filename:	Browse... Log/Display Only Errors Successes Configure											
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes		
Login	45	1138	34	3648	788.64	0.00%	18.0/second	0.00	0.00	9801.17		
Parse Data GRD	45	3802	178	11818	2098.40	0.00%	18.0/second	0.21	0.10	40388.13		
Mapping Directorate ..	45	34789	20	81210	20988.56	0.00%	18.0/second	0.21	0.00	40388.13		
Upload Data GRD	45	41777	220	86649	22626.11	0.00%	17.8/second	0.00	0.00	5603.0		
Parse Data GR	45	703613	408019	904009	108171.60	0.00%	18.0/second	0.00	481.32	67792.8		
Mapping Directorate ..	45	61749	41040	99621	12060.70	0.00%	18.0/second	0.20	0.00	40388.13		
Upload Data GR	45	38113	11480	133617	13313.57	0.00%	18.0/second	0.00	0.00	1275.0		
Parse Data FXA	45	1440997	1127927	1796629	187864.46	0.00%	14.0/second	0.00	74.77	79441.13		
Mapping Directorate ..	45	18206	27940	88178	12052.75	0.00%	18.0/second	0.20	0.00	40388.13		
Upload Data FXA	45	262177	100113	586111	101817.87	0.00%	18.0/second	0.00	0.00	3651.0		
Parse Data PD Adu	45	590178	404669	731118	87788.03	0.00%	17.6/second	0.16	0.17	118171.0		
Mapping Directorate ..	45	59117	37625	112640	12171.48	0.00%	18.0/second	0.19	0.00	40388.13		
Upload Data PD Adu	45	69428	40280	103844	12268.62	0.00%	18.0/second	0.00	0.00	1981.0		
Parse Data ZPS AUC	45	707286	421878	878131	98211.78	4.44%	15.0/second	0.79	49.07	100026.2		
Upload Data ZPS AUC	45	187911	77	252242	73313.86	0.00%	18.0/second	0.00	0.00	1001.0		
TOTAL	675	287966	20	1798629	400522.81	0.10%	3.1/second	2.27	194.32	45038.1		

Figure 7. Performance testing result with 15 concurrent users

A notable finding was the 4,44% error value recorded specifically by the Parse Data ZPS AUC module. This result was not identified as a system failure, but rather as evidence of successful exception handling, as the system was able to detect and reject anomalous variations that naturally occurred within the real dataset used during testing. This module-specific error rate indicates that the ZPS AUC table contained a relatively higher proportion of problematic data compared with other modules during the tested period, making the result a realistic reflection of actual operational data conditions. This error-handling mechanism preserved the integrity of the main database, which was confirmed by the subsequent Upload Data ZPS AUC module, where the error rate remained at 0,00%. The stability of this backend layer supports the performance of Monica's reporting dashboards because the data displayed and downloaded by users is sourced directly from the validated ETL processing pipeline. However, when users export dashboard reports to PPTX format, the system performs an additional client-side rendering step that captures the interface and charts directly within the user's browser. Therefore, the download speed may also depend on the user's device capability and network connection, which was identified as a direction for further performance testing in future system development.

These results indicate a trade-off between reliability and response time as concurrent load increases. While Monica maintained a low global error rate under both scenarios, the sharp increase in Parse Data FXA processing time under stress load, from 7.3 minutes with five concurrent users to approximately 24 minutes with fifteen, shows that the current architecture is not yet optimized for high-concurrency, CPU-intensive parsing tasks. In practical terms, this suggests that Monica performs reliably within its intended operational context, up to approximately five concurrent Admin users completing routine monthly uploads, but should not yet be considered suitable for scenarios requiring many users to process large Excel files simultaneously without an extended timeout. Rather than treating the low error rate alone as evidence of system stability, this bottleneck should be read as a defined capacity limit that identifies CPU-bound Excel parsing under concurrent load as the primary constraint on Monica's scalability, and represents a concrete direction for future optimization, such as asynchronous task queuing or horizontal scaling of the parsing service.

CONCLUSION

This research successfully designed and built a web-based Extract, Transform, Load automation system called Monica that fulfilled all of its stated objectives. The system provides automated data-cleansing and schema-validation features covering duplicate removal, date-format standardization, and missing-value handling, thereby improving the efficiency and standardization of asset and capitalization data preparation in the Asset Accounting Department of PT RS. It also implements a MySQL-based centralized repository that guarantees the consistency and traceability of historical data for audit and further-analysis purposes, and provides an integrated reporting dashboard covering the Asset Capitalization Report and the Monitoring Capex Report as a replacement for external Business Intelligence tools, so that the entire process from data preparation to report

presentation can be carried out within a single browser-accessible platform without additional installation. These achievements were reinforced by the cutover-phase testing results: Alpha Testing on 34 test cases achieved 100% success, Beta Testing produced a feasibility of 96.4% for Admin and 99.1% for User, both in the “Very Feasible” category, and Performance Testing showed error rates of 0.00% under normal load and 0.30% under high load with a stable throughput. Overall, Monica supports the reduction of manual data-preparation effort and the potential for human error, as reflected in respondents' evaluation and the automation of tasks previously performed manually, and establishes a single source of truth for the entire staff of the Asset Accounting Department, supporting a more efficient and structured monthly-report process. Performance testing further indicates that Monica operates reliably within its current intended scale of concurrent Admin users, while identifying Excel-parsing throughput under high concurrency as a specific limitation for future optimization. Future development may extend support to additional source-data formats such as CSV and JSON or direct connections to ERP systems such as SAP, add scheduled-ETL capability for fully automatic periodic loading, and adopt server-side rendering for dashboard-report generation to make report downloads faster and more consistent across devices and network conditions.

REFERENCES

- Agarwal, P., Singhal, A., & Garg, A. (2017). SDLC Model Selection Tool and Risk Incorporation. *International Journal of Computer Applications*, 172(10), 6–10. <https://doi.org/10.5120/ijca2017915143>
- Aliyah, Hartono, N., & Muin, A. A. (2024). Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. *Switch : Jurnal Sains Dan Teknologi Informasi*, 3(1), 84–100. <https://doi.org/10.62951/switch.v3i1.330>
- Basid, I. A. M., Islamiyah, N., Zuleika, R. A., Inka, A., Andarini, S., & Kusumasari, I. R. (2024). Peran Teknologi Informasi dalam Perencanaan dan Pengembangan Bisnis di Era Digital: Tantangan dan Peluang. *Economics And Business Management Journal*, 3(1).
- Fatha, M. K., Wati, S. F. A., Dewa, B. S., & Prasetyo, K. E. (2023). Peran Big Data Pada Intelijen Bisnis Sebagai Sistem Pendukung Keputusan (Systematic Literature Review). *Prosiding Seminar Nasional Teknologi dan Sistem Informasi*, 3(1), 318–326. <https://doi.org/10.33005/sitasi.v3i1.612>
- Firdaus, M., & Shedriko, S. (2024). Implementasi ETL dan Pengolahan Data Pelaporan Basel-III OJK di Bank Umum. *Seminar Nasional Riset dan Inovasi Teknologi*, 8(01). <https://doi.org/10.30998/semnasristek.v8i01.7131>
- Fitriawan, M. D. (2024). Analisis dan Visualisasi Data Penjualan di Toko Baju Sederhana Menggunakan Power BI (Pt. 783-793). *Jurnal Sistem Informasi*, 2(10).
- Hakky, M. K., Wirasmita, R. H., & Uska, M. Z. (2018). Pengembangan Media Pembelajaran Berbasis Android untuk Siswa Kelas X Pada Mata Pelajaran Sistem Operasi. *Jurnal Pendidikan Informatika*, 2(1), 24. <https://doi.org/10.29408/edumatic.v2i1.868>
- Hambling, B., & Goethem, P. van. (2013). *User Acceptance Testing: A step-by-step guide*. BCS Learning and Development Ltd.
- Hardika, B., Kurniawan, M. D., Adzka, M., Prastowiyono, D., Banyubasa, A., Wicaksono, A., & Nasir, M. (2024). Pengujian Blackbox Testing Website Garuda Farm Menggunakan Teknik Equivalence Partitioning. *Jurnal Kridatama Sains dan Teknologi*, 6(02), 740–753. <https://doi.org/10.53863/kst.v6i02.1420>
- Hidayat, A., Martini, R., Ristianingrum, A., Priatna, W. B., Indrawan, P., Manalu, D. S. T., Wiraguna, E., Dewi, H., Dewi, S. P., Pratama, A. J., Ratnawati, B., Putri, R. G. H., Manaf, I., Balqis, N., Hanafi, M. H., Ginting, K. J., Rizkialita, A., Darma, R. Y. B., Darmawan, M. D. M., & Siskandar, R. (2024). Predictions and Policies on Carbon Footprint Release Data at the College of Vocational Studies, IPB University (SV IPB) Based on the Contribution of Campus Operational Activities. *Environment and Ecology Research*, 12(1), 40–53. <https://doi.org/10.13189/eer.2024.120105>
- Hidayat, A. P., Wiyoto, W., Julio Pratama, A., Vibowo, H., Husen Santosa, S., & Siskandar, R. (2023). Fuzzy Analytical Hierarchy Process (AHP) Model for Chicken Egg Supply and Demand Management Strategies Through SAFCES Application Development. *E3S Web of Conferences*, 454, 03004. <https://doi.org/10.1051/e3sconf/202345403004>

- Hosen, M. S., Islam, R., Naeem, Z., Folorunso, O., Chu, T. S., Al, M. A., & Orunbon, N. O. (2024). Data-Driven Decision Making: Advanced Database Systems for Business Intelligence. *Nanotechnology Perceptions*, 20(S3). <https://doi.org/10.62441/nano-ntp.v20iS3.768>
- Irzaman, Suryana, Y., Pambudi, S., Widayanti, T., Jenie, R. P., Prastowo, B., Har, N. P., Rahmawaty, V., Dahrul, M., Aminullah, Kurniawan, A., Siskandar, R., Hardyanto, I., Iskandar, J., Nurdin, N. M., Ardidarma, A., Rahayu, S. K., & Alatas, H. (2022). Development of blood hemoglobin level early detection device based on a noninvasive optical platform. *Heliyon*, 8(11), e11260. <https://doi.org/10.1016/j.heliyon.2022.e11260>
- Kusuma, A. P., & Yufon, A. (2024). Analysis of User Acceptance Testing on A Shipping Application to Determine The Quality of The System. *Antivirus : Jurnal Ilmiah Teknik Informatika*, 18(2), 234–243. <https://doi.org/10.35457/antivirus.v18i2.4002>
- Listy, V., & Ilham. (2025). Revolusi Sistem Informasi Manajemen di Era AI dan Big Data Mengubah Cara Bisnis Bekerja. *Jurnal Sistem Informasi Dan Informatika*, 5(1). <https://doi.org/10.31294/simpatik.v5i1.7621>
- Manik, J. S. R., Yuaziva, A., Maulidya, I., Farah Fakhirah, S., & Siskandar, R. (2026). Digitalization of Forestry Museum Visiting Services Through the Implementation of a Reservation and Scheduling Platform. *Journal of Applied Science, Technology & Humanities*, 3(3), 1185–1217. <https://doi.org/10.62535/h1seca22>
- Maulidya, I., Salma, C. O. P., Ramadhani, K. A., Farizi, A., Beda, G. H. K., Wirawisesa, R. B. A., Wicaksono, A., & Nasir, M. (2024). Pengembangan Sistem Pencatatan Pesanan Berbasis Website dengan Pendekatan Equivalence Partitioning pada Tahap Pengujian Black Box. *Bina Insani ICT Journal*, 11(2), 154. <https://doi.org/10.51211/biict.v11i2.3233>
- Mindara, G. P., Sholihah, W., Novianty, I., Marcelita, F., Siskandar, R., Ariyanto, D., Setiawan, A., & Firdaus, N. R. (2025). Calibration of Dissolved Oxygen Sensors in IoT Systems for Water Quality Monitoring in Aquaculture. *Jurnal Fisika Dan Aplikasinya*, 10(3). <https://doi.org/doi.org/10.21009/SPEKTRA>
- Muku, I. D. G. D. B., Bayupati, I. P. A., & Susila, A. A. N. H. (2023). Pengembangan Sistem Business Intelligence Dalam Monitoring Performa Perusahaan Multi Company. *SINTECH (Science and Information Technology) Journal*, 6(2), 99–106. <https://doi.org/10.31598/sintechjournal.v6i2.1394>
- Muzahra, N., & Nasution, M. I. P. (2025). Strategi Untuk Menjaga Konsistensi dan Akurasi Data Dalam Pengambilan Keputusan. *Jurnal Ilmiah Nusantara*, 2(4). <https://doi.org/10.61722/jinu.v2i4.5023>
- Novianty, I., Ferdika, A., Sholihah, W., Siskandar, R., & Sari, I. P. (2019). Design of Portable Weather Station Using MQTT Protocol. *2019 2nd International Conference of Computer and Informatics Engineering (IC2IE)*, 199–202. <https://doi.org/10.1109/IC2IE47452.2019.8940893>
- Nurfadhilah, A., Pitriya, A., & Hidayat, W. (2025). Desain Dashboard Web untuk Pemantauan dan Pelaporan Status Produksi di PT. Care Spunbond. *Saturnus : Jurnal Teknologi Dan Sistem Informasi*, 3(4), 191–201. <https://doi.org/10.61132/saturnus.v3i4.1117>
- Nuryana, M. L., Ibrahim, T., & Arifudin, O. (2024). Implementasi dan Transformasi Sistem Informasi Manajemen Di Era Digital. *Jurnal Tahsinia*, 5(9). <https://doi.org/10.57171/jt.v5i9.614>
- Prasetia, I. P. W., & Kurniawan, I. N. H. (2021). Implementasi ETL (Extract, Transform, Load) pada Data warehouse Penjualan Menggunakan Tools Pentaho. *Information Technology Journal*, 2(1). <https://doi.org/10.38043/tiers.v2i1.2844>
- Putra, I. G. N. A. T., Mahendra, I. N. A., & Putra, I. M. S. (2020). Implementasi ETL Data Warehouse dengan Konsep Fitur Metadata dan Cleansing Data pada Toko Kue. *SISTEMASI*, 9(2), 274. <https://doi.org/10.32520/stmsi.v9i2.732>
- Putra, I. M. S., & Putra, D. K. T. A. (2019). Rancang Bangun Engine ETL Data Warehouse dengan Menggunakan Bahasa Python. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 3(2), 113–123. <https://doi.org/10.29207/resti.v3i2.872>
- Putra, I. N. A., Hartati, R. S., & Aryani, N. W. S. (2019). Optimasi Proses ETL dengan Metode Heuristik untuk Membangun Data Warehouse. *JST (Jurnal Sains Dan Teknologi)*, 8(1), 35–43. <https://doi.org/10.23887/jstundiksha.v8i1.16002>
- Santosa, S. H., Hidayat, A. P., & Siskandar, R. (2021). SAFEA application design on determining the optimal order quantity of chicken eggs based on fuzzy logic. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 10(4), 858. <https://doi.org/10.11591/ijai.v10.i4.pp858-871>

- Santosa, S. H., Prayudha Hidayat, A., Siskandar, R., & Rizkiriani, A. (2023). Production Scheduling Based on Smart Forecasting Model of Bottled Mineral Water Products. *E3S Web of Conferences*, 454, 03003. <https://doi.org/10.1051/e3sconf/202345403003>
- Sartika, I. E., & Aritonang, R. DP. (2025). Pengaruh Input Data yang Akurat, Input Data Secara Tepat Waktu, Kejelasan Data yang Diinput dan Kelengkapan Data yang Diinput pada Aplikasi Sakti Terhadap Kualitas Laporan Keuangan di Direktorat Jenderal Pendidikan Anak Usia Dini, Pendidikan Dasar dan Menengah. *Journal of Accounting and Finance Management*, 6(2). <https://doi.org/10.38035/jafm.v6i2.1941>
- Siskandar, R., Dio, F. C., & Irzaman. (2021). Application Of Ba0.5Sr0.5TiO3 (Bst) Film Doped With RuO2 (0%, 2%, 4% and 6%) On A Rice-Stalk Cutting Robot Model Based On A Line Follower With Hc-05 Bluetooth Control. *Biointerface Research in Applied Chemistry*, 12(2), 2138–2151. <https://doi.org/10.33263/BRIAC122.21382151>
- Siskandar, R., Hidayat, A., Martini, R., Ristianingrum, A., Manalu, D. S. T., Priatna, W. B., Indrawan, P., Wiraguna, E., Dewi, H., Dewi, S. P., Pratama, A. J., Ratnawati, B., Kharismatunnisaa, F., Darmawan, M. D. M., Nurfadillah, F., Sitompu, E. A., Sabilla, C. H., Assariy, M. F., & Zulqisthi, D. (2024). SoltarinE: Solar charging station eco friendly as a charging solution for electric farm machinery. *E3S Web of Conferences*, 577, 01011. <https://doi.org/10.1051/e3sconf/202457701011>
- Siskandar, R., Indrawan, N. A., Kusumah, B. R., Santosa, S. H., Irmansyah, I., & Irzaman, I. (2020). PENERAPAN REKAYASA MESIN SORTIR SEBAGAI PENENTU KEMATANGAN BUAH JERUK DAN TOMAT MERAH BERBASIS IMAGE PROCESSING. *Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering)*, 9(3), 222. <https://doi.org/10.23960/jtep-l.v9i3.222-236>
- Siskandar, R., & Kusumah, B. R. (2019). Control Device Engineering for Aquaponic Monitoring System. *Aquacultura Indonesiana*, 20(2), 72. <https://doi.org/10.21534/ai.v20i2.151>
- Siskandar, R., Mandang, T., Hermawan, W., & Irzaman, I. (2023a). Engineering of Information Monitoring System Sensor Reading Data Based on Smart Wireless using NVIDIA Jetson Nano and Arduino Mega on Agricultural Spraying Machines. *Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering)*, 12(4), 921. <https://doi.org/10.23960/jtep-l.v12i4.921-936>
- Siskandar, R., Mandang, T., Hermawan, W., & Irzaman, I. (2023b). Thin Film Potential Ba0.5Sr0.5TiO3 (BST) doped with RuO2 6% as a Light Detecting Sensor at Solar Tracker ALSINTAN System in Microcontroller-Based. *Biointerface Research in Applied Chemistry*, 13(6), 545. <https://doi.org/10.33263/BRIAC136.545>
- Siskandar, R., Santosa, S. H., Hidayat, A. P., Suwanda, B. S., Vibowo, H., Barokah, W. T., Habiburrahman, M., Faqihhudin, M. R., & Nerol, I. D. (2025). Pelatihan Pengembangan Alat Komunikasi Cerdas Keamanan Lingkungan Berbasis Smartphone di Karang Taruna Desa Pakapasan Girang, Kabupaten Kuningan, Jawa Barat. *Jurnal Pengabdian Masyarakat Bhinneka*, 3(4), 774–784. <https://doi.org/10.58266/jpmb.v3i4.259>
- Siskandar, R., Santosa, S. H., Wiyoto, W., Kusumah, B. R., & Hidayat, A. P. (2022). Control and Automation: Insmoaf (Integrated Smart Modern Agriculture and Fisheries) on The Greenhouse Model. *Jurnal Ilmu Pertanian Indonesia*, 27(1). <https://doi.org/10.18343/jipi.27.1.141>
- Siskandar, R., Wiyoto, W., Hendriana, A., Ekasari, J., Kusumah, B. R., Halim, G., & Nugraha, I. J. (2022). Automated Redox Monitoring System (ARMS): An Instrument for Measuring Dissolved Oxygen Levels Using a Potential Redox Sensor (ORP) in a Prototype of Shrimp Farming Pond with an Internet-Based Monitoring System. *Journal of Aquaculture and Fish Health*, 11(2), 238–246. <https://doi.org/10.20473/jafh.v11i2.31487>
- Siskandar, R., Wiyoto, W., Santosa, S. H., Hidayat, A. P., Kusumah, B. R., & Darmawan, M. D. M. (2023). Prediction of Freshwater Fish Disease Severity Based on Fuzzy Logic Approach, Arduino IDE and Proteus ISIS. *Universal Journal of Agricultural Research*, 11(6), 1089–1101. <https://doi.org/10.13189/ujar.2023.110616>
- Siskandar, R., Wiyoto, W., Santosa, S. H., Sari, J. E., Darmawangsa, G. M., Hidayat, A. P., Dardanella, D., & Kusumah, B. R. (2023). Potential Readings of Water Turbidity Values Based on Optical Sensors on Fish-Rearing Biofloc Media. *Photonics Letters of Poland*, 15(1), 1–3. <https://doi.org/10.4302/plp.v15i1.1176>
- Stephens, R. (2012). *Beginning Software Engineering* (1st ed.). Wiley. <https://doi.org/10.1002/9781119209515>

- Sudiantini, D., Naiwasha, A., Izzati, A., W, A. A., A, B. P., & Rindiani, C. (2023). Penggunaan Teknologi Pada Manajemen Sumber Daya Manusia Di Dalam Era Digital Sekarang. *Digital Bisnis: Jurnal Publikasi Ilmu Manajemen dan E-Commerce*, 2(2), 262–269. <https://doi.org/10.30640/digital.v2i2.1082>
- Syaifullah, M. H. (2024). Visualisasi Data Kebencanaan Kabupaten Jepara Menggunakan Microsoft Power BI. *Journal of Information System and Computer*, 4(1). <https://doi.org/10.34001/jister.v4i1.835>