COMPLETE
Journal Computers, Electronics, and Telecommunications
ISSN 2723-4371, E-ISSN 2723-5912

Article

Improving the Efficiency of Goods Transfer Instruction Management through the Implementation of a RESTful API Architecture (Case Study: PT Perkebunan Nusantara IV Regional IV Jambi)

Ahmad Nasukha[1], Aski Maya Partiwi[2], Jihan Nabillah[3] , Nadhif Pandya Supriyadi[4] and Ragit Dwi Saputra[5]

[1,2,3,4,5] Faculty of Science and Technology, Information Systems Study Program, Sulthan Thaha Saifuddin State Islamic University Jambi, Indonesia
* Correspondence: askimayaprtiwi@gmail.com.

Received: 19 May 2026; Revised: 2 June 2026; Accepted: 16 July 2026; Published: 20 July 2026.

Abstract: The administration of Goods Transfer Instructions at PT Perkebunan Nusantara IV Regional IV was previously managed through a semi-digital system consisting of a central spreadsheet accessible only to the Accounting/Admin division, while memorandum documents from Estates/Factories, Technical Division, 4OPH/4RHE, 4BSH, and 4AKN were sent separately through Google Drive without direct links to the spreadsheet. This condition caused coordination delays, the risk of misplaced documents, and limited transparency in document status tracking. This study aims to design, implement, and evaluate a web-based IPB management application using a RESTful API architecture. The system was developed using the SCRUM framework, React.js for the frontend, Node.js and Express.js for the backend, Prisma ORM, SQLite, JSON Web Token (JWT), and Role-Based Access Control (RBAC). The novelty of this study lies in the implementation of RESTful API architecture to digitize the IPB workflow in a state-owned plantation enterprise environment that requires tiered validation, audit trails, and real-time status monitoring. Black-box testing showed that all main scenarios functioned according to requirements. Endpoint testing using Postman showed response times of 4–96 ms for the main endpoints, while quantitative performance testing showed average response times of 18.80 ms for the root endpoint, 85.20 ms for login, 55.00 ms for the IPB list, and 32.10 ms for the IPB summary. Based on internship observations, the comparison between the previous semi-digital process and the new digital process showed that processing time decreased from 5–7 working days to less than 1 working day. These results indicate that RESTful API architecture is effective in improving the efficiency, transparency, and accountability of the IPB process.

Keywords: Goods Transfer Instruction; RESTful API; SCRUM; Postman; Audit trail.

1. Introduction

1.1 Background

Digital transformation has become essential for plantation companies to improve the accuracy, efficiency, and trans-parency of administrative processes [4, 10]. At PT Perkebunan Nusantara IV Regional IV Jambi, the Goods Transfer Instruction (IPB) process is used to control the transfer of assets between estate/factory units. Before the new sys-tem was developed, the IPB process was already semi-digital: data were recorded in a central spreadsheet by the Accounting/Admin division, while supporting memorandum documents were sent separately through Google Drive. The workflow began when Estates/Factories created issues and uploaded documents, followed by the

Technical Di-vision forwarding documents to 4OPH/4RHE, 4OPH forwarding documents to 4BSH or 4AKN, and finally the IPB document being issued by 4AKN/4BSH. The separation between the spreadsheet and Google Drive documents, without direct links between them, made document status difficult to monitor, supporting documents difficult to trace, and approval history difficult to audit comprehensively.

1.2 Problem Analysis

Based on observations of the existing process, four main problems were identified: coordination delays because data and documents were stored in different media, the risk of Google Drive documents being misplaced or difficult to locate, the absence of real-time status tracking for requesters, and difficulty in compiling document histories for audit purposes. The PIECES analysis show in Table 1 shows that information, control, efficiency, and service are critical aspects that need to be improved through a centralized digital system [6]. The Economy aspect was not included in the table because this study focuses on operational workflow efficiency, document traceability, and system accountability rather than a financial cost-benefit analysis

Table 1. Summary of PIECES Analysis for the Previous Semi-Digital IPB System.

Variable	Problem Description	Impact
Performance	Spreadsheet data and Google Drive documents were managed separately	Document coordination and verification be-came slower
Information	Document status was not visible in real time	Requesters had to ask for status updates through direct confirmation
Controls	Memorandum documents in Google Drive were not directly linked to the spreadsheet	Documents were difficult to trace and the audit trail was weak
Efficiency	The approval flow depended on a central spreadsheet and Drive-based file submission	Recapitulation and validation required more time
Service	Spreadsheet access was limited to Accounting/Admin	Estates and Technical Division could not in-dependently monitor status

1.3 Research Gap and Novelty

Previous studies have widely discussed RESTful APIs in academic systems, ordering systems, and general backend services, but have not specifically addressed IPB digitization in a state-owned plantation enterprise environment that requires tiered validation and document auditability [3, 8, 7]. The main gap in the previous system was the absence of an integrated platform that unified the central spreadsheet, memorandum documents from Google Drive, role-based authorization, request validation, status tracking, and automated audit trails. The novelty of this study is the design and evaluation of a RESTful API-based IPB application that centralizes the submission flow, direct file upload through a website, technical validation, admin approval, and activity history documentation in one system hosted directly on the PT Perkebunan Nusantara IV Regional IV server.

1.4 Rationale for Selecting RESTful API

RESTful API was selected because it supports frontend–backend separation, lightweight JSON-based communication, and easier integration with other systems [8, 7]. An API is considered RESTful when it satisfies the principles of client-server architecture, statelessness, cacheability, layered system, and uniform interface [11]. These principles fit the IPB system requirements because each request must carry authentication information, be processed through structured endpoints, be validated by middleware, and produce consistent responses. Compared with the use of a separated central

spreadsheet and Google Drive, RESTful API provides better access control, data validation, integrated file upload, status tracking, and measurable endpoint testing.

2. Materials and Methods

2.1 SCRUM Framework

The system was developed using the SCRUM framework as an implementation of the Agile mindset, rather than as phases of the Rational Unified Process (RUP) [2, 9]. SCRUM was applied through three roles (Product Owner, Scrum Master, and Developers), three artifacts (Product Backlog, Sprint Backlog, and Increment), five events (Sprint, Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective), and five values (Commitment, Focus, Openness, Respect, and Courage). A two-week sprint cycle was used so that changes in internal company requirements could be handled quickly.

2.2 Product Backlog

Functional requirements were formulated from the previous semi-digital IPB process and prioritized based on oper-ational urgency as shown in Table 2.

Table 2. Product Backlog of the IPB System.

ID	Requirement	Description	Prioritity
F-01	RBAC Authentication	Login based on Estate, Technical, and Admin roles	High
F-02	IPB Submission	Input form for asset metadata and document attachments	High
F-03	Technical Validation	Upload and verification of OPH/AKN documents	High
F-04	Admin Approval	Tiered approval/rejection with reasons	High
F-05	Audit Trail	History of status changes and user activities	High Medium

2.3 Supporting Modeling

UML was used as a supporting artifact to validate the IPB workflow, not as the main focus of the study. Modeling was conducted to identify the main actors, login flow, IPB submission, technical validation, and document approval. In response to reviewer feedback, the sequence diagram was prepared using separate boundary, control, and entity notations so that the roles of the interface, API gateway, middleware, controller, ORM, and database could be explicitly distinguished as shown in Figure 1. The main research focus remains on the RESTful API architecture, endpoint definitions, and API performance evaluation.

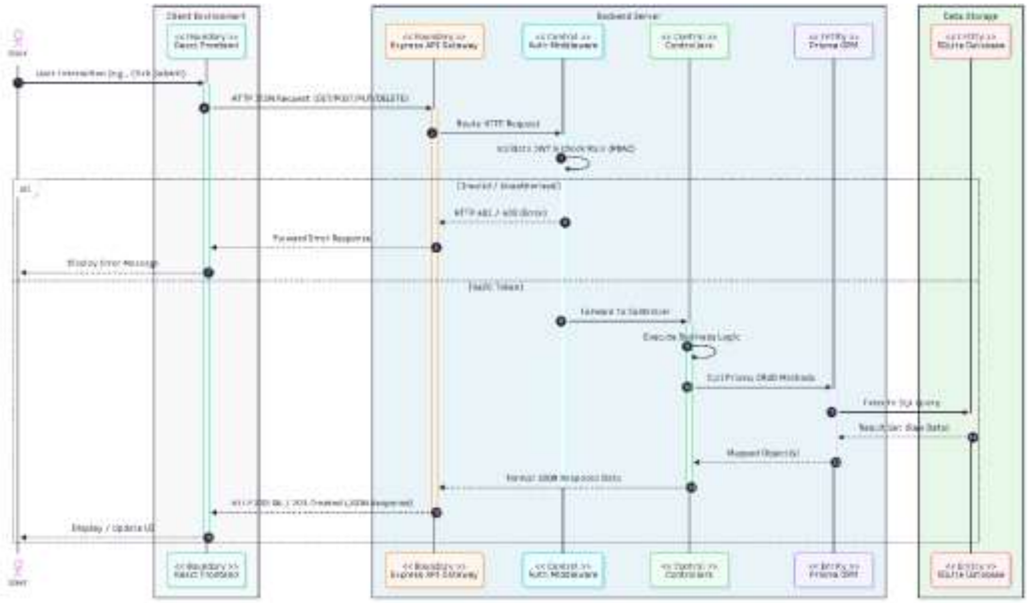

Figure 1. Standard Sequence Diagram

3. Results

3.1 Backend and Frontend Implementation

The backend was built using Node.js and Express.js to provide API services for the React.js frontend and has been hosted directly on the PT Perkebunan Nusantara IV Regional IV server. Authentication uses JWT, while authorization uses RBAC so that each role can only access endpoints according to its authority. Prisma ORM is used as the data access layer to SQLite. The React.js frontend is used for login, IPB monitoring dashboard, submission forms, direct memorandum document upload through the website, and the admin approval panel. The spreadsheet view of the previous semi-digital IPB system and the dashboard view of the new web-based IPB system are shown in Figure 2 and 3, respectively.

No	Bagian	Isu	Dokumen dari Kebun/Pabrik	Dokumen dari Bagian Teknis ke 4OPH/4RHE	Dokumen dari 4OPH ke 4BSH atau 4AKN	Dokumen IPB dari 4AKN/4BSH	Keterangan
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		[illegible]
49	4TAN	IPB Fertilizer Bayfolan 11% N;8% P2O5 LIQ dari Rambo Dua ke Danau Kembar	4KDR/4TAN/eM-107/IV/2026	4TAN/4OPH/eM-325/IV/2026	4OPH/4BSH/eM-94/IV/2026	4BSH/KOL/eM-50/IV/2026	SUDAH TERBIT IPB (DONE)
50	4TEP	IPB GASOLINEBIODIESEL CN 48,815-890KGM3 dari Bukit Kausar ke PKS Pengabuan		4TEP/4OPH/eM-224/IV/2026	4TEP/4AKN/eM-244/IV/2026	4BSH/KOL/eM-54/IV/2026, tanggal : 2026-04-17	sudah dilakukan transfer posting bulan maret

Figure 2. Spreadsheet View of the Previous Semi-Digital IPB System

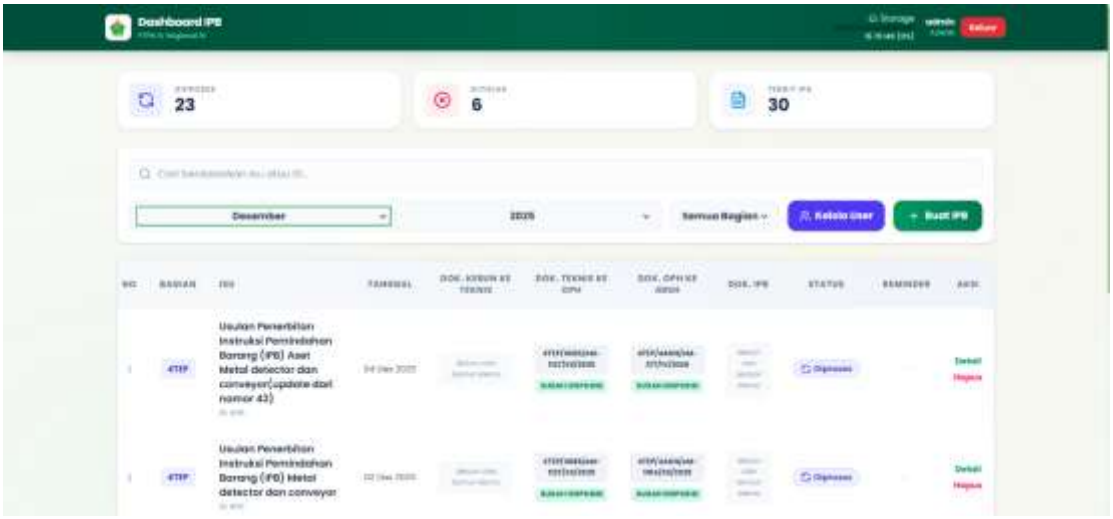

Figure 3. Dashboard View of the New Web-Based IPB System

3.2 System Architecture Diagram

In accordance with the research title, the main result of this study is the RESTful API architecture, while UML serves only as a supporting workflow representation. The system architecture diagram shows the JSON request flow from the React frontend to the Express API Gateway, JWT/RBAC validation through the Auth Middleware, processing by Business Logic/Controllers, data access through Prisma ORM, and storage in the SQLite database. Responses are returned to the client in JSON format through the same path. This layered separation demonstrates the principles of client-server architecture, stateless requests, layered system, and uniform interface, as shown in Figure 4.

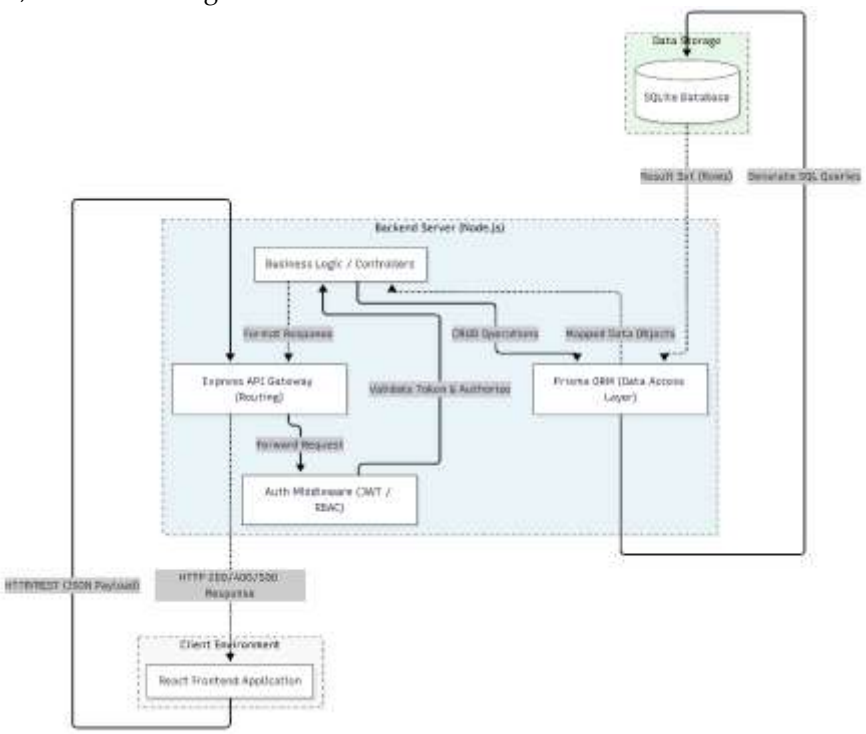

Figure 4. RESTful API-Based IPB System Architecture

3.3 API Endpoint Definition

The RESTful API uses structured endpoints to process authentication, IPB submissions, data retrieval, dashboard summaries, and data deletion according to access rights, as shown in Table 3.

Table 3. API Endpoint Functional Testing Results Using Postman.

No.	Endpoint	Method	Function	Main Request/Response	Status
1	/	GET	Backend health check	Response: API running message	200
2	/api/auth/register	POST	Registers a user account	Request: user data; Response: user created	201
3	/api/auth/login	POST	Validates credentials and issues JWT	Request: username/password; Response: JWT token	200
4	/api/auth/me	GET	Retrieves active user data	Request: bearer token; Response: user profile	200
5	/api/ipbs	POST	Creates a new IPB submission	Request: IPB form-data; Response: IPB ID	201
6	/api/ipbs	GET	Displays IPB list accord-ing to role	Response: data array and pagination	200
7	/api/ipbs/summary	GET	Displays dashboard summary	Response: total, in process, issued, rejected	200
8	/api/ipbs/:id	DELETE	Deletes IPB data according to access rights	Request: ID parameter; Response: success message	200

3.4 Quantitative Perfomance Testing

In addition to endpoint status testing, quantitative performance testing was conducted to observe response time stability, throughput, and error rate. Testing was performed using Postman through repeated request scenarios against the main endpoints. During this performance testing, the rate-limiting middleware was temporarily disable to prevent artificial request rejection and to measure endpoint response time, throughput, and error rate under controlled repeated-request scenarios. Therefore, the observed error rate of 0.00% reflect endpoint stability during testing without rate-limiter interference, as shown in Table 4.

Table 4. API Endpoint Quantitative Performance Testing Results.

Controller	Method	Endpoint	Avg.	Min	Max	Throughput	Error
Root	GET	/	18.80 ms	3 ms	215 ms	1012.15 req/sec	0.00%
authController	POST	/api/auth/login	85.20 ms	42 ms	312 ms	115.40 req/sec	0.00%
ipbController	GET	/api/ipbs	55.00 ms	6 ms	128 ms	346.47 req/sec	0.00%
ipbController	GET	/api/ipbs/summary	32.10 ms	12 ms	155 ms	560.85 req/sec	0.00%

The system still includes rate-limiting middleware as a protection mechanism for production use. However, stress testing with the limiter enable was not included in this performance testing scenario and is left for future evaluation as shown in Figure 5.

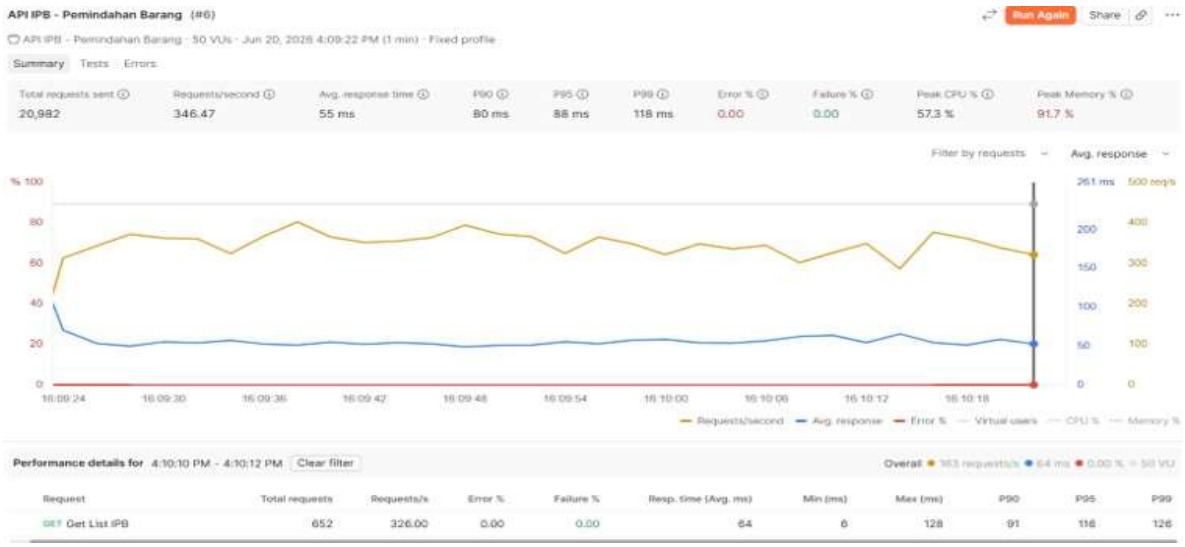

Figure 5. Postman Performance Test Result for GET List IPB Endpoint

3.5 Functional Testing

Black-box testing was conducted to ensure that the main features functioned according to requirements [5]. The tested scenarios included role-based login, document format validation, approval flow, and validation of unsaved changes, as shown in Table 5.

Table 5. Functional Test Result of the IPB System

ID	Scenario	Expected Result	Status
T-01	Role-Based Login	Users enter the dashboard according to their authority	Passed
T-02	Document Filter	The system rejects files other than PDF	Passed
T-03	Approval Flow	Status and audit trail are updated	Passed
T-04	Unsaved Change Validation	A confirmation dialog appears when data have not been saved	Passed

3.6 Efficiency Evaluation

The efficiency evaluation was based on general observations during the internship by comparing the previous semi-digital IPB process and the new digital IPB process during the trial period. The compared parameters included completion time, status tracking, document risk, and audit trail availability as shown in Table 6.

Table 6. Comparison between the Previous Semi-Digital IPB Process and the New System

Aspect	Previous Semi-Digital System	New System	Impact
Completion time	5–7 working days based on general internship observations	Less than 1 working day in the trial	Faster approval process
Status tracking	Not real-time and dependent on Accounting/Admin	Real-time through the dashboard	Increased transparency
Document risk	Google Drive memorandum documents were separated from the spreadsheet	Direct file upload through the website	More integrated archiving
Activity audit	Approval history was difficult to trace	Recorded through audit trails	Increased accountability

4. Discussion

The results show that the implementation of RESTful API architecture can reduce bureaucratic bottlenecks because requests from the frontend are processed through standardized endpoints, validated by JWT/RBAC, and recorded in an audit trail. Compared with the previous semi-digital approach, which separated the central spreadsheet from memorandum documents in Google Drive, this system is superior because it provides a centralized data flow, integrated document upload, real-time status, role-based access control, and rate-limiting mechanism prepared for production protection. Compared with RESTful API studies in ordering or academic systems, this study contributes to the context of plantation asset administration, which requires tiered validation and documented decision histories [3, 8].

The practical implication of this architecture is that each estate/factory unit can submit and monitor IPB without sending memorandum documents separately through Google Drive. The technical implication is that the system can be further developed for ERP integration because the frontend, backend, and database have been separated into clear layers. The limitations of this study include the use of SQLite, which is not optimal for high concurrency, the limited number of evaluation samples, and load testing that was still conducted under limited scenarios and therefore does not yet represent large-scale production load testing. Although the system has been hosted on the PT Perkebunan Nusantara IV Regional IV server, load testing was still limited; therefore, further evaluation under larger access loads is required to assess system stability, scalability, and performance when accessed by many users simultaneously. Future development is recommended to use PostgreSQL, integrate with the central ERP system, and provide a mobile application for field staff.

5. Conclusion

This study successfully designed, implemented, and evaluated a RESTful API-based IPB management web application at PT Perkebunan Nusantara IV Regional IV Jambi. The system transforms the semi-digital process based on a central spreadsheet and separated Google Drive documents into a centralized, authorized, and auditable digital workflow. Postman testing showed that the main endpoints returned appropriate HTTP status codes, low response times, stable throughput, and an error rate of 0.00%. The efficiency evaluation also showed a decrease in processing time from 5–7 working days to less than 1 working day. Therefore, RESTful API

architecture is feasible for improving the efficiency, transparency, and accountability of the IPB process.

References

1. PT Perkebunan Nusantara IV. (2024). Strengthened by Solid Integration, Towards a World-Class Company.
2. Arnomo, S. A., & Kurniawan, D. E. (2024). Agile Scrum Method in the Development of an Inventory Control System. 169–177.
3. Brema Adinta, Winarsih, S. N. (2024). Implementation of RESTful API in a Food Ordering and Delivery System. 5(2), 122–129.
4. Cholik, C. A. (2021). Development of Information and Communication Technology/ICT in Various Fields. 2(2), 39–46.
5. Desyani, T., Mulyati, S., Kurnianto, E., Afifah, N., Nur, S., & Fauziah, I. (2022). Black-box Testing Using Equiv-alence Partitioning Techniques on the Best Employee Selection System Application. 5(2), 110–114.
6. Hulu, L., & Nurlelah, E. (2026). Performance Analysis of the Inventory Information System at PT SBID Using the PIECES Framework. 14(1), 13–21.
7. Khoiri, M. A., Pratama, N. W., & Rangkuti, M. Y. (2026). Implementation of RESTful API as a Backend Service for Web Applications Using Laravel and MySQL. 3(11), 2875–2878.
8. Mohammad Akmaluddin Novianto, S. M. (2022). Analysis and Implementation of RESTful API for the Devel-opment of Academic Information Systems in Higher Education. 8(1), 47–61.
9. Ramadhan, C., Senubekti, M. A., & Amalia, D. (2025). Application of Agile Methodology in Software Devel-opment. 3(2).
10. Raza, E., Sabaruddin, L. O., & Komala, A. L. (2021). Benefits and Impacts of Logistics Digitalization in the Industry 4.0 Era. 4(1), 49–63.
11. Shofyan, F., & Shita, R. T. (2024). Implementation of RESTful API Web Services with Personal Access Token Authentication and Algorithms. 12, 108–114.