Design and Implementation of a Backend System and DevOps Workflow for Interactive Learning Applications

Alfian Bisma Daniswara Sirwenda*[1], Sasono Wibowo[2], Indra Gamayanto[3], Slamet Ikhvan Nurhana Rifki[4], Ivan Ismanto[5], Michael Christ Kurniawan[6]
[1,2,3,4,5]Faculty of computer science, Information system department, Dian Nuswantoro University, Semarang
alfianbisma2021@gmail.com [1], sasono.wibowo@dsn.dinus.ac.id [2], indra.gamayanto@dsn.dinus.ac.id [3], 112202206882@mhs.dinus.ac.id [4], 112202206763@mhs.dinus.ac.id [5], 112202206752@mhs.dinus.ac.id [6]

Article Info

Article history:

Received 2025-09-24
Revised 2025-10-31
Accepted 2025-11-08

Keyword:

Backend,
DevOps,
Gamification,
English Language Learning,
Rapid Application Development,
CI/CD Pipeline.

ABSTRACT

English language learning in Indonesia faces significant challenges, including limited vocabulary retention, poor pronunciation, and passive learning methods. The EngVenture application was developed to address these issues by integrating gamification principles with interactive English learning environments. This study aims to design and implement a backend system and DevOps workflow that ensure optimal performance, security, and stability for gamification-based learning applications. The Rapid Application Development (RAD) method was employed, comprising requirements planning, user design, construction, and cutover phases. System requirements were identified through a validated questionnaire (Cronbach's $\alpha = 0.89$) distributed to 101 respondents from diverse backgrounds. Results indicated that users prioritized data security (90.1%), system speed (91.1%), and secure authentication (69.3%) as critical factors. Based on these findings, a RESTful API-based backend was designed and integrated with Docker, Jenkins, and Nginx, incorporating security features such as JWT authentication, API key validation, and SSL/TLS encryption. Quantitative evaluation over a 20-day period demonstrated significant improvements: 85% faster deployment time (6.23→1.48 minutes), 43.4% reduction in error rate (211→138 errors), 95.7% build success rate, stable API response time (~160ms) under load testing with 1,000 concurrent requests, and near-zero downtime (<5 minutes). This research demonstrates that the integration of structured backend architecture and automated DevOps practices significantly enhances system reliability, deployment efficiency, and user satisfaction in educational technology applications such as EngVenture.

I. INTRODUCTION

English has become the world's primary lingua franca, functioning as a global medium of communication across education, business, diplomacy, and technology[1][2][3]. Mastery of English supports not only linguistic competence but also intellectual and professional competitiveness in the international arena. In Indonesia, English education has been integrated into the curriculum from an early stage to prepare students for the global era[1], yet challenges remain in optimizing learning outcomes. According to Syandri[2], English for Specific Purposes (ESP) also plays an essential role in vocational and higher education contexts, aligning linguistic skills with professional needs. Consequently, improving English language instruction requires innovative strategies that strengthen learner motivation, adaptability, and technological literacy. Qualified teachers and effective pedagogical designs are crucial to ensure that language learning becomes more relevant and impactful for students in the 21st century.

Technological advancement has profoundly transformed the educational landscape by facilitating creativity, accessibility, and collaboration in learning[4][5][6] Safitri and Tari[4] emphasize that gamification-based learning enhances motivation and language proficiency, especially when supported by interactive digital tools. Similarly, Purnasari and Sadewo[5] note that digital transformation provides equitable access to quality learning even in remote areas. Sulistyarini et al.[6] further highlight that digital literacy and the use of learning media improve teachers' pedagogical competence in digital learning environments. In the context of language learning, technology facilitates diverse instructional media videos, online quizzes, and real-time feedback that encourage students to engage actively in learning activities. It enables flexible learning environments where students can practice listening, reading, and speaking skills dynamically. Thus, educational technology functions not merely as a supplementary tool but as an integral part of effective and modern English pedagogy.

Despite early English education, many Indonesian students continue to face challenges in mastering essential language skills such as pronunciation, reading comprehension, and vocabulary acquisition[7], [8]. Khairum et al.[7] argue that the use of technology in elementary education can enhance language learning quality but still requires adaptive methods that match student behavior. Common issues include word mispronunciation, literal reading, and limited vocabulary retention, which hinder learning progress[8]. Gamification-based learning has been identified as an effective approach to address these problems by integrating game mechanics such as challenges, rewards, and progress tracking into learning activities[9][10]. Demonstrate that game-based vocabulary learning in elementary schools significantly improves learner engagement and retention. Similarly,[10] highlight that platforms such as *Kahoot* and *Quizizz* encourage competitive yet collaborative learning environments[10]. Games not only provide enjoyment but also stimulate active participation, as supported by Yufenti Oktafiah[8], who found that creative educational games can increase students' motivation and curiosity in learning English.

While gamification has proven effective in enhancing engagement, research on its integration with robust backend and DevOps infrastructure remains limited. The *EngVenture* application represents an innovative effort in Indonesia that applies gamification principles to interactive English learning, integrating dynamic features and real-time feedback. However, most previous studies on educational DevOps have focused primarily on deployment automation or integration pipelines, without incorporating quantitative performance analysis or user-centric evaluations[11][12]. Fernandez-Gauna et al. [9] and Offerman et al.[13] emphasize the pedagogical benefits of DevOps in collaborative coding but note the lack of structured metrics in educational contexts. Similarly, Jayakody and Wijayanayake[11] and Azad and Hyrynsalmi[14] identify critical success factors for DevOps adoption but do not examine its role in learning applications requiring high reliability and security. This research gap underscores the need for integrating measurable DevOps workflows in educational systems, particularly those supporting gamified and interactive environments where uptime, security, and scalability directly affect the learning experience.

This study contributes by proposing an integrated backend and DevOps architecture tailored for interactive language learning applications. The architecture combines RESTful API, JWT-based authentication, and containerized CI/CD pipelines using Docker, Jenkins, and Nginx—following the structured system development approach inspired by the Rapid Application Development (RAD) methodology[15]–[19]. This design ensures efficient iteration, modular scalability, and continuous integration aligned with agile educational requirements. Quantitative results demonstrate 85% faster deployment, 43.4% fewer errors, and a 95.7% build success rate compared to traditional systems. Multi-layered security (API Key, JWT, SSL/TLS) ensures data integrity, addressing 90.1% of users who expressed high concern for system protection. A validated questionnaire (Cronbach's $\alpha = 0.89$, $n = 101$) confirms that users prioritize system speed (91.1%) and authentication reliability (69.3%). Furthermore, the system supports real-time interaction with <160 ms response time and zero-downtime deployment, essential for gamified learning environments. These findings extend current literature by demonstrating how DevOps integration can enhance performance, reliability, and pedagogical efficiency in educational technology systems.

II. METHOD

Figure 2. 1 EngVenture Application Research Process

A. Identifying Problems

In the initial stage, the main focus is finding relevant, real-world problems to research. Researchers examine phenomena or needs that remain unresolved or for which no solution has been found. This stage ensures that the research has a strong foundation and is relevant to the needs of the community, organization, or field of science.

B. Formulating Research Questions and Objectives

Next, the research questions and objectives are determined. These objectives and questions will serve as benchmarks for the success of the research.

C. Literature Review

At this stage, researchers collect and study literature related to their research subject. Sources may include books, scientific journals, previous research reports, and other reliable sources. The main objective of this research is to gain a comprehensive understanding of the theories, methods, results, and objectives of previous studies. Another objective is to identify research gaps that this research will fill.

D. Rapid Application Development (RAD)

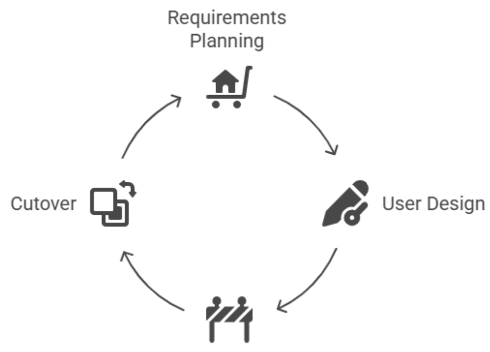

Figure 2. *2* RAD (Rapid Application Development)[20]

The Rapid Application Development (RAD) method was chosen due to its iterative and adaptive nature, allowing continuous collaboration between developers and users during the backend design and DevOps integration stages. RAD emphasizes both accuracy and development speed through repeated prototyping and user feedback in every phase of the process. This method integrates the structured principles of the Waterfall model with the flexibility of iterative development, focusing on efficiency and responsiveness to user needs.[12]. The use of RAD in this study ensures that the backend and CI/CD workflow can be refined quickly based on testing results and user validation, leading to a more reliable and user-centered system.

In research, RAD is used to create application prototypes that solve previously identified problems. This approach involves rapid, iterative software design, development, and testing to ensure the application meets user needs. The stages of the RAD method that are implemented include:

1) *Requirements Planning*

The main focus at this stage is identifying the users' system requirements and defining the project scope[21]. Developers gather information about user requirements for the upcoming information system. This information is then used to create the features, workflows, and services users expect[15].

2) *User Design*

At this stage of the research, the backend design has been developed according to the requirements. This includes creating the API service process flow and proposed system architecture, as well as designing the database table structure to support the application features[16]. This phase is carried out to meet the technical design requirements of back-end users and allow them to provide input so the system is built according to their needs[17].

3) *Contructions*

The construction phase begins the process of creating the previously designed system. At this stage, developers write program code to transform the design into an actual application according to the plan. Users remain actively involved during this phase, providing input and suggesting changes or improvements to the system. Additionally, progress reports are continuously submitted to ensure that the system meets the requirements[18].

4) *Cutover*

The cutover phase is a stage in the implementation of DevOps that ensures the EngVenture application backend can operate stably in the production environment[19]. The main activities at this stage include automatic deployment through the Jenkins pipeline, service management in containers with Docker, and server configuration with Nginx and SSL to ensure secure access. By the end of the cutover phase, the backend will be fully developed and successfully migrated to the server infrastructure, allowing application users to access and use it properly.

E. Data Collection

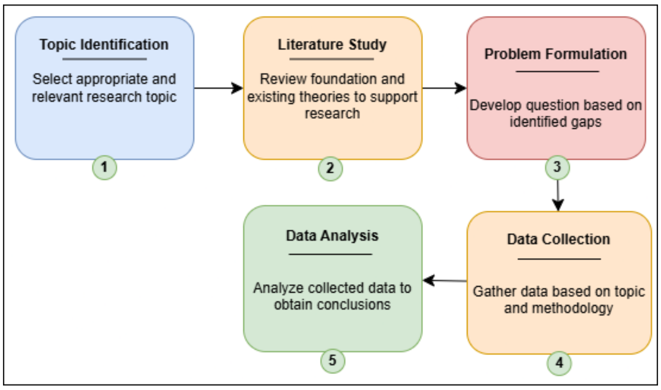

Figure 2. *3* Data Collection

The following image illustrates the steps of the research methodology used to design and develop the backend system and the DevOps process for the EngVenture application at the IntSys Research Lab. The staircase shape of the diagram indicates that the research process is carried out gradually, systematically, and structurally.

1) Topic Identification

The research began with the identification of a need for a modern, adaptive learning system, especially in the context of gamified language learning. The EngVenture application was designed to address these challenges by prioritizing speed of access, data security, and reliable backend system integration.

2) Literature Study

A review of literature and previous research was conducted to understand technical and methodological approaches to backend and DevOps development. Topics covered in the analyzed literature included CI/CD-based DevOps management and user data security in digital education systems.

3) Questionnaire Formulation

To gather user requirements and expectations, a questionnaire was developed. The questionnaire included questions about important backend system features, security expectations, access speed expectations, and user preferences for authentication methods.

4) Questionnaire Distribution

The questionnaire was distributed to 101 respondents, most of whom were students. As these respondents were representative of the target users of the EngVenture application, their responses were invaluable in designing a fit-for-purpose system.

TABLE I
RESPONDENT PROFILE

Category	Count	Percentage
Pelajar	26	25.7%
Mahasiswa	33	32.7%
Karyawan	8	7.9%
Wirausaha	26	25,7%
Ibu Rumah Tangga	2	2%
Guru	2	2%
Belum Bekerja	2	2%
Dagang	1	1%
TNI	1	1%
Total	101	100%

A total of 101 respondents were purposively selected to represent EngVenture's target users. The sample included 26 high school students (25.7%) and 33 university students (32.7%), accounting for 58.4% of participants actively learning English. In addition, 26 working professionals (25.7%) from various sectors using English for communication and 16 others (15.9%)—including teachers, entrepreneurs, and self-employed learners—were also represented. Demographically, respondents consisted of 39 individuals aged over 25 years (38.6%), 34 aged 20–25 years (33.7%), 27 aged 15–20 years (26.7%), and 1 under 15 years (1%). The group comprised 66 males (65.3%) and 35 females (34.7%), mostly educated at the high school to bachelor's degree level. This diverse composition ensures that the system requirements reflect the needs of a broad range of English learners, enhancing the generalizability of the study findings for digital language learning applications.

TABLE III
SUMMARY OF QUESTIONNAIRE ITEMS

No	Indicator	Item Statement	Scale
1	System Speed	The application should load quickly (<3 seconds)	Likert 1–5
2	System Speed	API response time should be fast for smooth interaction	Likert 1–5
3	System Speed	No lag during gameplay or learning activities	Likert 1–5
4	Data Security	My personal data must be encrypted and protected	Likert 1–5
5	Data Security	Secure login with password protection is essential	Likert 1–5
6	Data Security	System should prevent unauthorized access	Likert 1–5
7	Authentication	Email and password login is my preferred method	Likert 1–5
8	Authentication	Two-factor authentication adds security	Likert 1–5
9	Reliability	Application should be accessible 24/7 without downtime	Likert 1–5
10	Reliability	System should auto-recover from errors	Likert 1–5

The questionnaire used in this study was validated through expert judgment by two information systems lecturers specializing in software engineering and educational technology, who assessed each item for content validity, clarity, and relevance to the research objectives, with items receiving less than 80% agreement being revised or removed.

Reliability testing using Cronbach's Alpha yielded a coefficient of 0.89, surpassing the 0.70 threshold, indicating strong internal consistency. Subscale reliability scores were also high, including System Speed (α = 0.85), Data Security (α = 0.88), Authentication (α = 0.82), and System Reliability (α = 0.87). These results confirm that the questionnaire is both valid and reliable for evaluating user requirements in the EngVenture backend system.

5) Data Collection

The responses were collected and compiled for analysis. Quantitative data obtained from various diagrams, such as gender, age, institution, and technical user requirements (important features, authentication, and downtime tolerance), formed the basis for designing the backend system and DevOps pipeline.

6) Data Analysis

The data was analyzed to determine users' actual needs. For instance, most users considered login and security to be essential features, and the maximum tolerable downtime was under five minutes. These results informed the development of a backend system focused on reliability, efficiency, and security, as well as DevOps practices supporting high availability and zero-downtime deployment.

III. RESULT AND DISCUSSIONS

A. User Requirements

In the early stages of implementing the Rapid Application Development (RAD) model, data was collected by sending 101 respondents from various backgrounds and educational levels Google Forms questionnaires. The survey results were used to determine user requirements, such as application access speed, data security, and important features. These findings then formed the basis for the backend and DevOps workflow and were discussed further in the Results and Discussion section to ensure the designed system met user requirements.

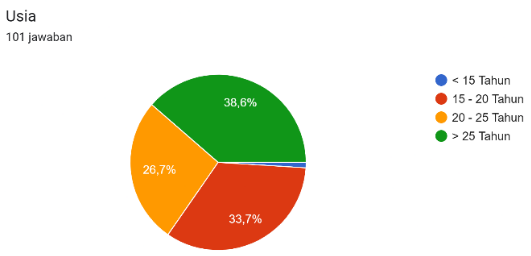

Figure 3. *1* Age Distribution of Respondents

The respondents came from various age groups. Most were over 25 years old: 39 respondents (38.6%). The second largest group was 20–25 years old (34 respondents, 33.7%), followed by 15–20 years old (27 respondents, 26.7%). Only one respondent (1%) was under 15 years old.

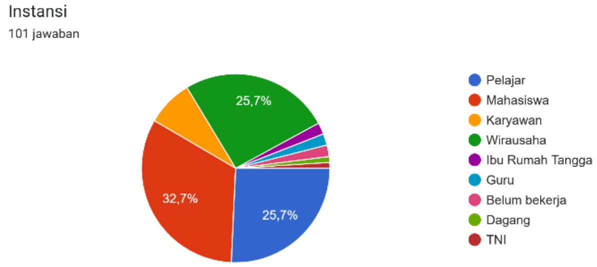

Figure 3. *2* Background of Respondents' Institutions or Occupations

The majority of respondents came from various institutions. The largest group was made up of 33 respondents (32.7%), followed by students and employees with 26 respondents each (25.7%). Fewer respondents came from the following groups: entrepreneurs, housewives, teachers, the unemployed, traders, and military personnel.

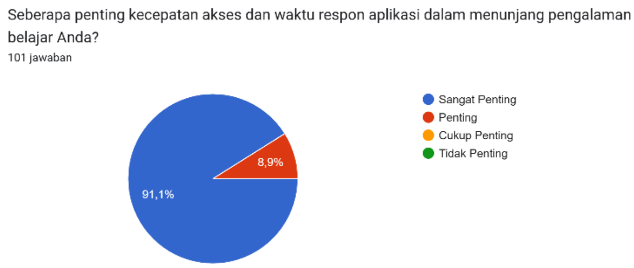

Figure 3. *3* Level of Importance of Application Access Speed and Response in Supporting The Learning Experience

Overall, 91.1% of respondents rated application speed and response time as *very important* to their learning experience, while 8.9% considered it *important*, and none viewed it as less significant. These findings emphasize that EngVenture backend development must prioritize efficient data processing, effective request management, and optimized server architecture to maintain low latency, especially under high traffic loads. Given that gamification-based applications rely on fast, seamless interactions to sustain engagement, system sluggishness can quickly reduce user interest. To address this, EngVenture employs a lightweight RESTful API, an Nginx reverse proxy for efficient traffic distribution, and Docker containers to maintain stable, high-performance operation across all environments.

Figure 3. *4* Level of User Concern for Data Security and Privacy

Data security and privacy are key concerns among users, with 90.1% expressing a *very high level of concern* about the protection of their personal data, 7.9% indicating concern, and only 1% showing minimal worry. This highlights the critical need for strong security measures within the EngVenture backend system. As a digital learning platform that stores sensitive user information such as identities and activity records, EngVenture applies best DevOps security practices, including encryption, token-based authentication using JSON Web Tokens (JWT. Additionally, all communications are secured via SSL/HTTPS to ensure data confidentiality and maintain user trust in the platform's integrity and safety

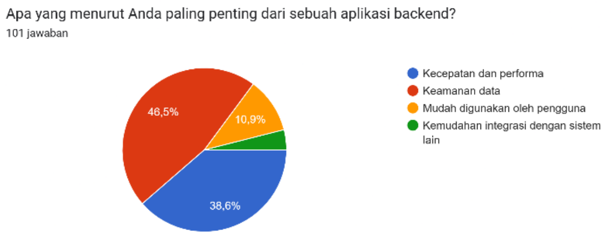

Figure 3. *5* Most Important Elements According to Users of Backend Applications

As illustrated in the figure, the survey results show that users prioritize both system speed and data security in backend applications, with 46.5% considering data security the most important factor and 38.6% emphasizing speed and performance, while 10.9% valued ease of use and only 4% mentioned ease of integration. These findings highlight that EngVenture users demand a secure yet high-performing system. To address this, the backend is designed with strong security mechanisms such as password encryption, input validation, JSON Web Tokens (JWT), and HTTPS-only connections, while performance is enhanced through query optimization, asynchronous processing, and load balancing. This combination ensures that EngVenture maintains both robust data protection and responsive system performance, aligning with user expectations for reliability and speed.

Figure 3. 6 Application Features Vonsidered Very Important by Users

Based on the survey results, 69.3% of respondents identified login and security as the most important features, followed by activity history or important data (10.9%) and notifications and alerts (8.9%), while only 1% emphasized aspects such as recognizable display, data security, and ease of use. These findings indicate that EngVenture users prioritize secure and reliable system access above all else, making the implementation of multi-factor authentication and real-time activity logging essential. At the same time, notification features that inform users about level progression, achievements, and learning updates support the gamification aspect of the application. Overall, the strong emphasis on security demonstrates that safe access is a fundamental prerequisite before users engage with other features.

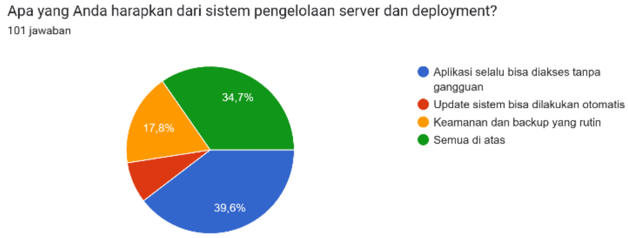

Figure 3. 7 Expectations for Server Management and Deployment Systems Infrastructure and Expectations for DevOps Systems

The findings indicate that 39.6% of respondents prioritize uninterrupted access, 34.7% value all service aspects (continuous availability, automatic updates, security, and regular backups), 17.8% focus on security and backups, and 7.9% emphasize automatic updates. These results highlight that service availability is the primary concern in user perception. To address this, EngVenture's backend should implement load balancing, container auto-restart (Docker/Kubernetes), zero-downtime deployment, and automated rollback within the CI/CD pipeline. Additionally, automated updates and backups can be ensured through scheduled cron jobs, incremental backup systems, and integrated notifications using Jenkins or GitHub Actions.

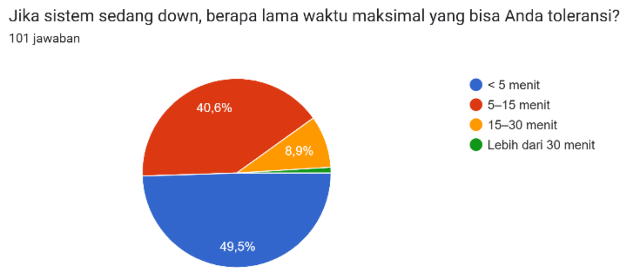

Figure 3. 8 System Security and User Authentication Preferences

Most respondents (49.5%) stated that the maximum acceptable downtime is less than five minutes, while 40% tolerated 5–15 minutes, 8.9% allowed 15–30 minutes, and only 1% accepted more than 30 minutes. These results emphasize the critical need for implementing robust DevOps practices, including failover systems, redundancy mechanisms, and fast, reliable CI/CD pipelines to ensure continuous service availability and minimize downtime.

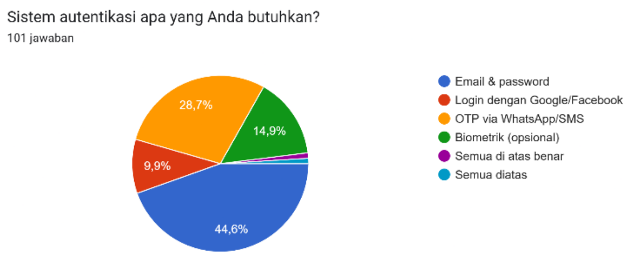

Figure 3. 9 User Expectations for Server Management and Deployment Systems

The majority of respondents (44.6%) prefer password and email authentication as their primary login method. This method is considered the most efficient and is widely used. Additionally, 28.7% of respondents prefer the OTP (one-time password) system via WhatsApp or SMS because it provides an additional level of security and is practical. Only 9.9% of respondents use third-party accounts, such as Google or Facebook, to log in. Additionally, 14.9% of respondents choose biometrics as an optional method. While most respondents use conventional authentication methods, many support dual authentication methods. This indicates that systems supporting multi-factor authentication (MFA) need to be developed. This is particularly important for addressing user security concerns, especially for applications like EngVenture that handle user activity history and personal data.

B. User Design

During the user design stage, the backend system is designed according to the previously analyzed user requirements to describe how the service will run. This design process involves data structure modeling and identifying the relationships between entities in the database using Entity Relationship Diagram (ERD). ERD illustrate the relationships between primary tables, including users, sessions, quiz questions, streaks, hearts, and leaderboards.

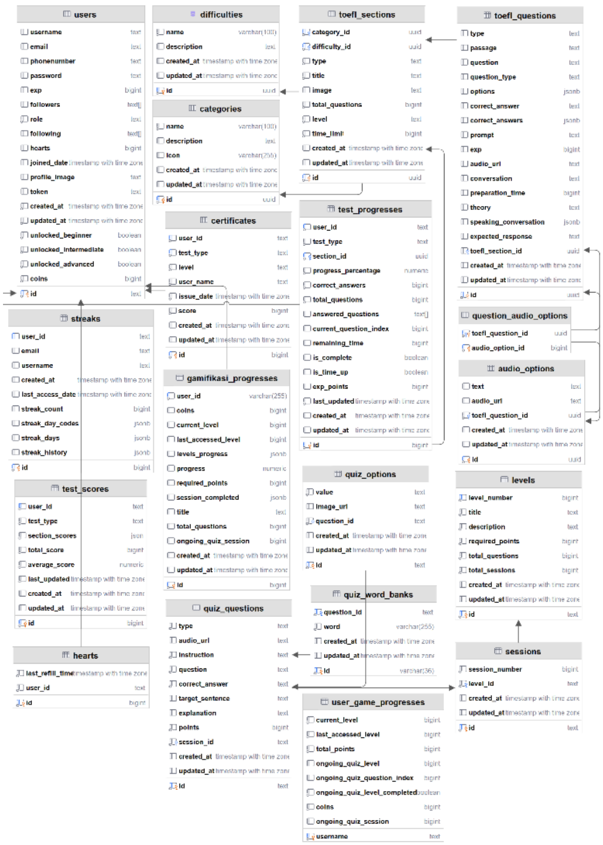

Figure 3. 10 Entity Relationship Diagram (ERD)

Figure 3.10 illustrates the EngVenture database schema, which unifies user management, learning progress, gamification, and TOEFL test data within a normalized relational structure. Core entities such as users, sessions, and toefl_questions are connected to supporting tables like categories, difficulties, and levels to ensure integrity and scalability. Learning and gamification progress are tracked through test_progresses, gamifikasi_progresses, and streaks, while certificates and test_scores record assessment results. Question entities, including quiz_questions, quiz_options, and question_audio_options, manage varied item types and multimedia content. This schema effectively supports adaptive learning, progress monitoring, and consistent data flow across EngVenture's integrated backend–frontend ecosystem.

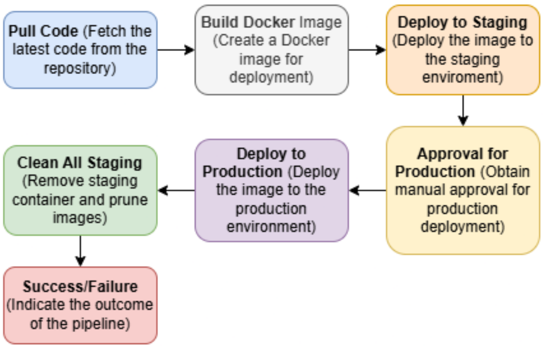

Figure 3. 11 CI/CD Pipeline Stages

Figure 3.11 illustrates the CI/CD process. It begins with retrieving the latest code from the repository and building a Docker image as a release-ready application package. Next, the image is deployed to the staging environment for functional testing, integration, and configuration verification. If the test results meet the standards, the process moves on to manual approval before deployment to the production environment. Once approved, the same image is promoted to production to ensure consistency with the staging environment. Then, the pipeline cleans up the staging environment by stopping temporary containers and deleting unused images. The final stage displays the process's final status as either successful or failed, which serves as feedback for further improvements.

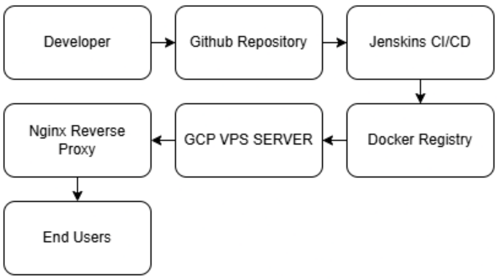

Figure 3. 12 Deployment Flowchart

Figure 3.12 depicts the application deployment flow using an integrated CI/CD pipeline. Developers commit code to GitHub, triggering Jenkins to automate compilation, testing, and Docker image creation. The generated image—containing the application and its dependencies—is stored in a Docker Registry to ensure consistent environments across development, testing, and production. The image is then deployed to a VPS server, where the application runs in isolated containers managed through Nginx reverse proxy for secure and efficient traffic handling. This automated process enables faster, more reliable, and consistent deployments, minimizing manual errors while maintaining continuous service availability.

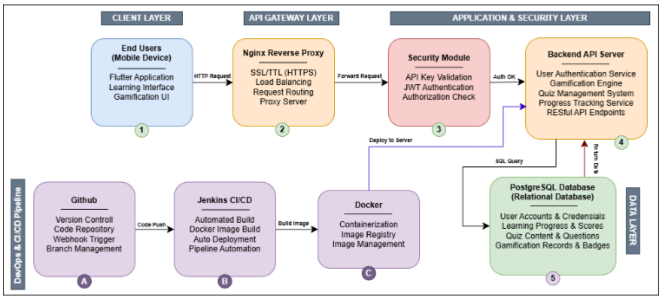

Figure 3. 13 Engventure Arsitektur system workflow

Figure 3.13 illustrates the EngVenture system architecture, which integrates the client, gateway, application, and data layers through an automated DevOps pipeline. The Flutter-based mobile client provides a gamified learning interface, while the Nginx reverse proxy in the API gateway manages SSL/TLS encryption, load balancing, and secure request routing. The backend API service, supported by a security module implementing JWT-based authentication and authorization, handles user management, gamification logic, and RESTful API operations connected to a PostgreSQL database for storing user, progress, and assessment data. Meanwhile, the DevOps pipeline—comprising GitHub for version control, Jenkins for automated CI/CD processes, and Docker for containerized deployment—ensures system consistency, scalability, and rapid, reliable updates aligned with modern software delivery practices.

C. Construction

The construction phase focused on developing the EngVenture backend at the IntSys Research Lab, including server-side coding, database design, and API service development to connect the application with the server. Testing using Postman and Swagger confirmed the proper functioning of core endpoints such as user authentication and data management, ensuring the backend system was fully operational before deployment to the production environment.

```
ENDPOINT =
https://asynfxy.web.id/golang_api/api/auth/login
(POST)

REQUEST =
{"usernameOrEmail": "testing123","password":
"12345678"}

RESPONSE =
{"user":{"id":"testing123_
1","username":"testing123","email":"testing123
@gmail.com","phonenumber":"232853728","exp":0,"fo
llowers":[],"role":"user","following":[],"hearts"
:0,"joined_date":"2025-10-27T08:00:42.652052+
07:00","profile_image":{"String":"","Valid":false
},"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e
yJ1c2VyX2lkIjoidGVzdGluZzEyM18xIiwidXNlcm5hbWUiOi
J0ZXN0aW5nMTIzIiwiaWF0IjoxNzYxNTMxNDA3fQ._
95bYoQikckCNYgUWKDnwey5T73UD7dvsI62BwL6Cso","coin
s":0,"unlockedBeginner":true,"unlockedIntermediat
e":false,"unlockedAdvanced":false,"created_at":"2
025-10-27T15:00:44.07669+
07:00","updated_at":"2025-10-27T15:00:44.07669+
07:00"},"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpX
VCJ9.eyJ1c2VyX2lkIjoidGVzdGluZzEyM18xIiwidXNlcm5h
bWUiOiJ0ZXN0aW5nMTIzIiwiaWF0IjoxNzYxNTMxNDA3fQ._
95bYoQikckCNYgUWKDnwey5T73UD7dvsI62BwL6Cso"}
```

Figure 3. 14 Endpoint Login

The image shows the results of testing the following endpoint using Postman during the construction phase: https://asynfxy.web.id/golang_api/api/auth/login. The test results show a 200 OK response, indicating that the request was processed successfully and complete user data was generated, along with a JWT token that will be used to maintain the authentication session for subsequent requests. The backend implementation also uses API keys to restrict service access and ensures that user passwords are stored in an encrypted form to maintain data confidentiality. These results demonstrate that the login service functions as designed and has adequate security protections for use in a production environment.

TABLE IIIII
SYSTEM ENVIRONMENT AND TECHNICAL SPECIFICATIONS

Component	Specification
Server	Google Cloud Platform (VPS)
CPU	2 vCPU
RAM	8 GB
Storage	200 GB SSD
Operating System	Ubuntu 22.04 LTS
Containerization	Docker
CI/CD Tool	Jenkins
Web Server	Nginx
Database	PostgreSQL
SSL Certificate	Let's Encrypt

The performance of the backend system and DevOps workflow was quantitatively evaluated to measure the impact of CI/CD integration on deployment efficiency, system reliability, and API performance stability. Key metrics included deployment time (from code commit to production), API response time (average latency of critical endpoints), error rate (percentage of failed requests), and system downtime (total service unavailability). Data were collected from Jenkins build logs, Nginx access logs, and load testing using the Postman Collection Runner with 1,000 concurrent requests. The evaluation was conducted in two phases before DevOps integration (manual deployment baseline) and after full CI/CD implementation (automated pipeline) over a 20-day testing period to ensure consistency and reliability of the results.

D. Cutover

The cutover phase is part of the DevOps implementation that ensures the EngVenture Application backend can run stably in the production environment. The main activities in this phase include the automated deployment process through the Jenkins pipeline, service management in containers using Docker, and server configuration with Nginx and SSL to ensure access security. With the cutover phase, the backend is not only completed but also successfully migrated to the server infrastructure so that it can be reliably accessed and used by application users.

Figure 3. 15 Jenkins Pipeline Display with Successful Build Status

The image shows the results of implementing a pipeline on Jenkins for the EngVenture application's backend project at the IntSys Research Lab. The build process was successfully executed, as indicated by the green status icon on build number #78. This indicates that the code integration from the repository was processed without errors. Jenkins' implementation is an essential component of the Continuous Integration/Continuous Deployment (CI/CD) concept because it enables the automated testing and distribution of every code change to the server. If the build process is ongoing or not yet accepted for deployment to the production environment, the status indicator is a running bar, and the green color turns gray. Thus, the backend system can be consistently maintained, minimizing manual errors and supporting more efficient DevOps practices during the cutover phase.

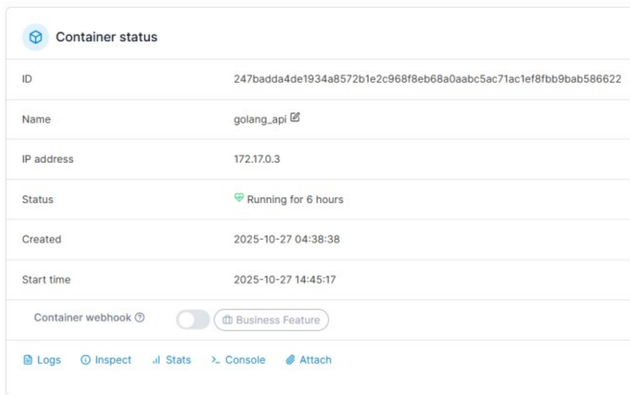

Figure 3. 16 A Golang Docker Container is Running on The Server

Figure 3.16 shows the container resulting from compiling the EngVenture application backend in a staging environment with Docker. The figure shows that the golang_api_staging container runs from the asyncrons/kp-golang:latest image by executing the ./main command as the application's starting point. The container remains active for several minutes with an "up" status and maps host port 7001 to internal port 8080 so the backend service can be accessed according to the specified network configuration. This indicates that the deployment process at the staging stage was successful and that the backend system is ready for testing before moving to the production environment.

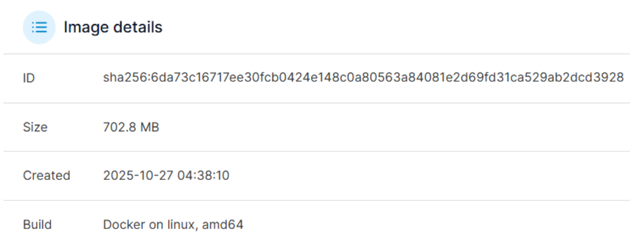

Figure 3. 17 Successfully Built A Golang Docker Image for The Backend

Figure 3.17 shows the details of the Docker image used to run the container. The asyncrons/kp-golang:latest image, which was generated about an hour before the screenshot was taken, is approximately 702.8 MB in size. The unique ID, sha256:6da73c16717ee30fcb0424e148c0a80563a84081e2d69fd31ca529ab2dcd3928, marks the latest version of the image resulting from the build process. This information proves that the pipeline successfully built the backend image, which was then used in the active container. Together, Figures 3.16 and 3.17 show that the Continuous Integration/Continuous Deployment (CI/CD) flow has been running consistently because the built image can be used directly for testing in an isolated, controlled staging environment.

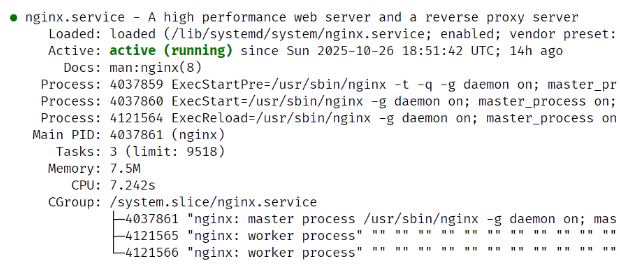

Figure 3. 18 Nginx Service Status

The configuration testing results indicate that the nginx.conf file is valid, with no syntax errors detected, and the service status shows that Nginx is currently active (running). This confirms that the server is successfully operating as a reverse proxy and is ready to handle incoming client requests. The system logs also display several reload activities, which typically occur after configuration updates to ensure that new settings are applied without interrupting server availability. Under these conditions, the server is able to manage network traffic effectively while maintaining service reliability and security.

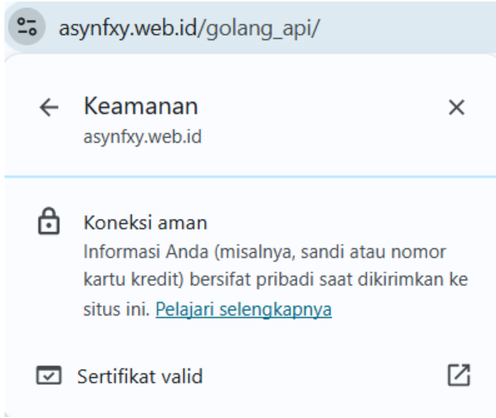

Figure 3. 19 SSL Security Display on The Domain asynfxy.web.id

As shown in the image, the domain asynfxy.web.id has been successfully secured with a valid SSL certificate. A padlock icon and a secure connection message in the browser indicate that all data exchanged between users and servers is encrypted using the HTTPS protocol. This ensures that sensitive information, such as credentials or personal data, cannot easily be accessed by third parties during transmission. Validation of the certificate by a trusted authority further strengthens security, ensuring that the connection between the EngVenture application and its users is secure and compliant with DevOps best practices during the cutover phase.

TABLE IV
SYSTEM PERFORMANCE COMPARISON BEFORE AND AFTER DEVOPS INTEGRATION

Parameter	Before DevOps Integration	After DevOps Integration	Improvement
Average Deployment Time	6.23 min	1.48 min	↓85% (faster)
API Response Time	160 ms	159 ms	≈0% (stable)
Error Rate	211	138	↓43.4% (lower)
Downtime	< 5 min	< 5 min (not detected)	0% (stable)
Build Success Rate	78% (manual, sering gagal dependency)	95.7 % (otomatis Jenkins)	↑ 22.7 %

Table IV compares system performance metrics before and after DevOps integration, evaluated from 22 September to 29 October 2025 using Jenkins build logs, Nginx access logs, and Docker container records. Each metric was calculated quantitatively based on system monitoring data using the following formulas: API Response Time = Σ(Server Response Time) / Number of Requests, Error Rate = (Number of 5xx Errors / Total Requests) × 100%, Downtime = Σ(Duration of Service Unavailability), Build Success Rate = (Successful Builds / Total Builds) × 100%, and Improvement = (Before Value – After Value) / Before Value × 100%. The Average Deployment Time was obtained from start–finish timestamps in the Jenkins pipeline, API Response Time from load testing (Postman Collection Runner), Error Rate from HTTP 5xx logs in Nginx and Docker, Downtime from service interruption and container uptime data, and Build Success Rate from Jenkins automation reports. This quantitative approach provides a measurable and empirical evaluation of system efficiency, reliability, and stability improvements after DevOps integration.

E. Discussion and Impact on User Satisfaction

The implementation of DevOps practices in the EngVenture backend system has significantly improved both technical performance and user experience without compromising essential functionality. Through Jenkins-based automation and Docker containerization, the system achieved an 85% reduction in average deployment time and a 43.4% decrease in error rate, indicating that the automated CI/CD pipeline effectively minimizes human error, accelerates release cycles, and maintains service continuity with near-zero downtime. From the user perspective, these technical improvements directly enhance satisfaction by enabling faster feature updates, reducing API response latency, and ensuring greater system stability. Moreover, enhanced reliability and multi-layered security mechanisms (API Key, JWT, and SSL/TLS) have strengthened user trust, addressing the 90.1% of respondents who prioritized data protection and accessibility. Overall, the integration of DevOps not only optimized operational efficiency but also fostered a secure, responsive, and dependable learning environment that supports continuous improvement in the EngVenture platform.

IV. CONCLUSIONS

This research successfully designed and implemented a backend system and DevOps workflow for EngVenture, a gamification-based English learning platform. Requirements analysis through a validated questionnaire (Cronbach's α = 0.89, n = 101) identified data security (90.1%), system speed (91.1%), and secure authentication (69.3%) as primary user priorities. The system integrates a RESTful API with JWT authentication, API key validation, and SSL/TLS encryption, deployed through an automated CI/CD pipeline using Docker, Jenkins, and Nginx.

Quantitative evaluation over a 20-day period demonstrated significant improvements: 85% faster deployment (6.23→1.48 minutes), 43.4% error reduction (211→138 errors), 95.7% build success rate, and stable API response time (~160ms) under load testing with 1,000 concurrent requests. These results directly address empirically identified user requirements and validate the effectiveness of integrating structured DevOps practices in educational technology systems.

The study has two limitations: prototype-stage testing over a limited period (20 days) and a sample size (n=101) that may not fully represent Indonesia's diverse English learning population. Future research should conduct large-scale production testing, implement Kubernetes for advanced orchestration, integrate AI-based personalized learning, and establish centralized log aggregation systems (Docker, Nginx, Jenkins, application logs) for real-time monitoring, automated alerting, and proactive maintenance to enhance long-term system reliability and observability.

ACKNOWLEDGMENTS

We would like to express our gratitude to Insys Research Lab for funding and providing full support for this research from start to finish.

REFERENCES

[1] N. Maru'ao, "Penerapan Pembelajaran Inovatif Dalam Meningkatkan Pembelajaran Bahasa Inggris," *Univ. Dharmawangsa*, vol. 14, no. 2, pp. 221–230, 2020.

[2] G. Syandri, "Pembelajaran Bahasa Inggris Dalam Konteks English for Spesific Purpose (Esp)," vol. 10, no. 1, pp. 19–28, 2023.

[3] S. P. Byslina Maduwu, "Pentingnya Pembelajaran Bahasa Inggris Di Sekolah," *Angew. Chemie Int. Ed. 6(11), 951–952.*, vol. 4, no. 1, p. 2017, 2016.

[4] N. P. D. Safitri and N. Tari, "Persepsi dan Pengalaman Mahasiswa Menggunakan Gamifikasi untuk Meningkatkan Motivasi dan Keterampilan Bahasa Inggris di Era Digital," *J. Simki Pedagog.*, vol. 7, no. 2, pp. 501–514, 2024, doi: 10.29407/jsp.v7i2.690.

[5] P. D. Purnasari and Y. D. Sadewo, "Strategi Pembelajaran Pendidikan Dasar di Perbatasan Pada Era Digital," *J. Basicedu*, vol. 5, no. 5, pp. 3089–3100, 2021, doi: 10.31004/basicedu.v5i5.1218.

[6] S. F. Sulistyarini, Winda, "Pengaruh Pemahaman Literasi Digital Dan Pemanfaatan Media Pembelajaran Terhadap Kompetensi Pedagogik Guru Era Digital Learning," *Educ. Learn. Innov.*, vol. 1, no. 2, pp. 98–116, 2021, doi: 10.46229/elia.v2i1.

[7] S. Y. Khairum, Adrias, and A. P. Zulkarnaini, "Peran teknologi dalam meningkatkan kualitas pembelajaran Bahasa Inggris di jenjang sekolah dasar," *Pendas J. Ilm. Pendidik. Dasar*, vol. 10, no. 1, pp. 573–586, 2025.

[8] D. L. R. Yufenti Oktafiah, "Game Sebagai Media Pembelajaran Kreatif Dan Menarik Untuk Meningkatkan Minat /Motivasi Belajar Bahasa Inggris Pada Anak Sekolah Dasar," *J-Abdi J. Pengabdi. Kpd. Masy.*, vol. 3, no. 6, pp. 781–792, 2023, [Online]. Available: https://doi.org/10.53625/jabdi.v3i5.6594

[9] B. Fernandez-Gauna, N. Rojo, and M. Graña, "Automatic feedback and assessment of team-coding assignments in a DevOps context," *Int. J. Educ. Technol. High. Educ.*, vol. 20, no. 1, 2023, doi: 10.1186/s41239-023-00386-6.

[10] M. L. Pedra, M. F. da Silva, and L. G. Azevedo, “DevOps Adoption: Eight Emergent Perspectives,” 2021, [Online]. Available: http://arxiv.org/abs/2109.09601

[11] J. A. V. M. K. Jayakody and W. M. J. I. Wijayanayake, “Critical success factors for DevOps adoption in information systems development,” *Int. J. Inf. Syst. Proj. Manag.*, vol. 11, no. 3, pp. 60–82, 2023, doi: 10.12821/ijispm110304.

[12] S. N. Bakri and M. I. P. Nasution, “Penerapan Metodologi Rekayasa Perangkat Lunak untuk Efisiensi Pengembangan Sistem,” *JSITIK J. Sist. Inf. dan Teknol. Inf. Komput.*, vol. 3, no. 1, pp. 53–66, 2024, doi: 10.53624/jsitik.v3i1.542.

[13] T. Offerman, R. Blinde, C. J. Stettina, and J. Visser, “A Study of Adoption and Effects of DevOps Practices,” *2022 IEEE 28th Int. Conf. Eng. Technol. Innov. ICE/ITMC 2022 31st Int. Assoc. Manag. Technol. IAMOT 2022 Jt. Conf. - Proc.*, 2022, doi: 10.1109/ICE/ITMC-IAMOT55089.2022.10033313.

[14] N. Azad and S. Hyrynsalmi, “DevOps critical success factors — A systematic literature review,” *Inf. Softw. Technol.*, vol. 157, no. April 2022, p. 107150, 2023, doi: 10.1016/j.infsof.2023.107150.

[15] C. C. Siburian, A. Andriani, and C. B. Dewa, “Aplikasi Mypresent Untuk Pengelolaan Data Presensi Karyawan Dengan Metode Rad,” *J. Inf. Syst. Manag.*, vol. 5, no. 2, pp. 219–226, 2024, doi: 10.24076/joism.2024v5i2.1435.

[16] D. Silvi Purnia, “Implementasi metode RAD pada Rancang Aplikasi BAN-SOS Ter Distribusi Berbasis Mobile,” *IJCIT (Indonesian J. Comput. Inf. Technol.*, vol. 3, no. 1, pp. 71–79, 2018.

[17] Y. Yanuardi, L. Azhari, A. A. J. Sinlae, and A. D. Alexander, “Pengembangan Sistem Pengaduan Layanan Masyarakat Menggunakan Metode Rapid Application Development (RAD),” *J-Intech*, vol. 12, no. 1, pp. 36–48, 2024, doi: 10.32664/j-intech.v12i1.1201.

[18] I. Roery, S. L. Mufreni, and E. P. Silmina, “Sistem Informasi Klinik Berbasis Web Dengan Menggunakan Metode Rapid Application Development (RAD),” *Pros. Semin. Nas. Penelit. dan Pengabdi. Kpd. Masy.*, vol. 3, no. 22, pp. 76–87, 2025.

[19] F. Al Ikhsan Ramadhan and B. Maula Sulthon, “Perancangan Sistem Administrasi Barang Berbasis Web Menggunakan Model Pengembangan Sistem RAD,” *Media Online*, vol. 4, No.1, no. 1, pp. 31–42, 2023, [Online]. Available: https://djournals.com/resolusi

[20] Nurman Hidayat and Kusuma Hati, “Penerapan Metode Rapid Application Development (RAD) dalam Rancang Bangun Sistem Informasi Rapor Online (SIRALINE),” *J. Sist. Inf.*, vol. 10, no. 1, pp. 8–17, 2021, doi: 10.51998/jsi.v10i1.352.

[21] A. Sah, S. D. Saraswati, N. Heriyani, and M. Badaruddin, “Pengembangan Sistem Pengeloaan Bimbingan Belajar Menggunakan Pendekatan Rapid Application Development,” vol. 4, no. 1, pp. 271–283, 2024, doi: 10.47065/comforch.v4i1.1409.