KOMPUTASI: JURNAL ILMIAH ILMU KOMPUTER DAN MATEMATIKA
VOL. 23 (1) (2026), 89-99 p-ISSN: 693-7554, e-ISSN:2654-3990

Design and Implementation Evaluation of a Teaching Assistant Information System Using Laravel Filament and Extreme Programming

Nasrul[1], **Henry Saptono**[2], **Edi Wibowo**[3], **David Wahyu Wismanindra**[4]

[1,2] Department of Informatics Engineering, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, West Java, 16451, Indonesia
[3] Department of Information Systems, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, West Java, 16451, Indonesia
[4] Department of Digital Business, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, West Java, 16451, Indonesia

Abstract

The management of teaching assistants at Sekolah Tinggi Teknologi Terpadu Nurul Fikri (STT-NF) has traditionally relied on manual processes, resulting in time inefficiencies, data inaccuracies, and limited information integration. To address these issues, a web-based Teaching Assistant Information System was developed using the Laravel Framework with Filament as the administrative interface and PostgreSQL as the database management system. The system supports teaching assistant recruitment, class and assistant scheduling, and automated honorarium calculation. The development process applies the Extreme Programming (XP) methodology to enable rapid iterations and continuous user feedback. Functional validation using Black Box Testing shows that all system features operate correctly, achieving a 100% test success rate. A comparative evaluation indicates that administrative processing time was noticeably reduced based on administrative staff feedback and data consistency improved compared to the previous manual approach. This study contributes by demonstrating the practical adaptation of XP in academic administrative systems and providing empirical evidence of efficiency improvement in teaching assistant management.

Keywords: *Extreme Programming; Filament; Laravel; Teaching Assistant Information System; Web-Based System*

1. Introduction

Sekolah Tinggi Teknologi Terpadu Nurul Fikri (STT-NF) is a higher education institution committed to integrating the development of information technology with Islamic values, competence, and strong character building. To support the quality of the learning process, STT-NF involves teaching assistants who assist lecturers in various academic activities, including class management, material preparation, and student evaluation [1]. Teaching assistants play a crucial role in supporting academic activities, particularly in practicum-based courses where their presence contributes significantly to the effectiveness and continuity of the teaching and learning process [2].

Along with the increasing number of teaching assistants and the growing complexity of their responsibilities, the manual management approach currently implemented at STT-NF has become inadequate. Several issues have been identified, such as inaccurate teaching assistant data, difficulties in tracking schedules, time-consuming honorarium calculations, and limited support for performance evaluation. The recruitment process of teaching assistants often requires considerable time due to the absence of an integrated digital system. Furthermore, class and assistant scheduling is prone to time conflicts, while honorarium calculations conducted manually increase the risk of errors and delays. These challenges indicate that manual management methods are no longer sufficient to meet institutional needs effectively.

Based on these conditions, a systematic and integrated solution is required to manage teaching assistant activities more efficiently and accurately. This research aims to develop a Teaching Assistant Information System as a digital solution to support the administration and management of teaching assistants at STT-NF.

*Corresponding author. *E-mail address:* nasrul@nurulfikri.ac.id
Received: 29 December 2025, Accepted: 30 January 2026 and available online 31 January 2026
DOI: https://doi.org/10.33751/komputasi.v19i2.5260

An information system is defined as a collection of interrelated components, including hardware and software, designed to collect, store, process, and manage data into meaningful information that supports organizational activities and decision-making processes [3].

Despite the availability of various academic and teaching assistant management systems, most prior studies focus primarily on feature implementation without providing empirical evaluation of development methodology effectiveness or post-deployment impact. Furthermore, limited attention has been given to adaptive agile practices tailored to academic administrative environments. This research addresses these gaps by integrating Extreme Programming with iterative evaluation and by presenting a comparative analysis between manual and automated management processes.

The Teaching Assistant Information System is developed using the Laravel Framework, an open-source PHP-based web application framework that is widely used due to its flexibility, security features, and developer-friendly architecture [4]. Laravel supports the implementation of the Model-View-Controller (MVC) design pattern, which separates business logic, data handling, and user interface components, thereby improving system maintainability and scalability [5]. In addition, the administrative interface of the system is developed using Filament, an admin panel package designed to simplify the development and management of Laravel-based applications.

For data management, this system utilizes PostgreSQL as the database management system. PostgreSQL is an open-source Object-Relational Database Management System (ORDBMS) that supports Structured Query Language (SQL) and provides advanced features such as data integrity, reliability, and extensibility [6]. These characteristics make PostgreSQL suitable for managing academic and administrative data in higher education institutions.

The development process of this system applies the Extreme Programming (XP) methodology, an agile software development approach that emphasizes iterative development, continuous user involvement, and rapid adaptation to changing requirements. The novelty of this research lies not only in the integration of Extreme Programming with modern web development technologies such as Laravel, Filament, and PostgreSQL, but also in the empirical evaluation of its impact on administrative efficiency through comparative analysis between manual and automated management processes. This approach provides practical evidence of how XP can be effectively adapted to academic administrative systems in higher education institutions.

2. Methods

This research adopts an experimental approach consisting of data collection methods and a software development method using Extreme Programming (XP). The applied methodology is designed to ensure that the developed Teaching Assistant Information System meets user requirements, operates efficiently, and can adapt to changes during the development process.

2.1. Data Collection Methods

Data collection was conducted to obtain accurate information related to the management of teaching assistants at STT-NF. The data collection methods used in this study are as follows:

a. Observation

Observation was carried out to collect data and information by directly examining the teaching assistant management processes at STT-NF. This method involved observing activities related to recruitment, scheduling, and honorarium calculation. In addition to direct observation, indirect observation was also conducted through the examination of documents, records, and other available information relevant to teaching assistant management. This approach allowed researchers to understand existing workflows and identify system requirements comprehensively.

b. Literature Study

The literature study method was conducted by reviewing and analyzing relevant references related to teaching assistant management, information system development, and current technologies in web-based applications. The information obtained from journals, books, and previous studies served as a theoretical foundation for designing, developing, and implementing an effective and efficient Teaching Assistant Information System in accordance with the needs of STT-NF.

2.2. Research Stages

After completing the data collection process, the next step was system development using the Extreme Programming (XP) methodology. XP is an agile software development method that focuses on short development cycles, continuous feedback, and rapid responses to changing user requirements [7]. XP was selected over other agile methodologies such as Scrum and Kanban due to its strong emphasis on short development cycles, continuous user involvement, and frequent testing. In the context of academic

administration systems, where requirements frequently evolve based on policy and user feedback, XP provides greater flexibility and responsiveness. Previous studies have also reported that XP is effective in small-to-medium scale information systems requiring rapid adaptation and close stakeholder collaboration. This methodological choice also enables measurable evaluation across development iterations, including functional completeness and testing outcomes, which are reported in the Results and Evaluation section. The application of the XP methodology in this study is illustrated in Figure 1.

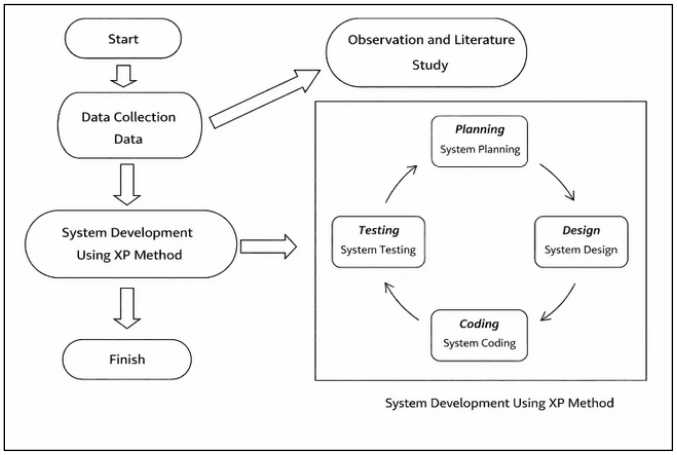

Figure 1. Extreme Programming Development Method

The stages of system development using the Extreme Programming methodology are described as follows:

a. Planning

The planning phase involves preparing a detailed development plan, including project objectives, system scope, and estimations of time and resources. At this stage, the development team identifies and prioritizes the main features required in the Teaching Assistant Information System to fulfill user needs and institutional requirements [8].

b. Design

The design phase includes intensive discussions among team members to design the system architecture and define system features. The design of the Teaching Assistant Information System is developed by considering user requirements, technical constraints, and integration with existing systems. This phase aims to produce a flexible and scalable system design that supports efficient development and future enhancements [9].

c. Coding

During the coding phase, the development team implements system features based on the agreed design. The coding process is conducted collaboratively using pair programming, where developers work in pairs to improve code quality, reduce errors, and enhance knowledge sharing among team members [10].

d. Testing

Testing is an integral part of each iteration in Extreme Programming. The development team performs unit testing, integration testing, and user acceptance testing continuously to detect and resolve system errors early [11]. In this study, system testing is conducted using the Black Box Testing method. Black Box Testing focuses on evaluating system functionality from the user's perspective to ensure that the system operates according to specified requirements and produces the expected outputs [12]. While this study focuses on functional validation through Black Box Testing, usability and performance evaluations are identified as potential extensions for future research.

3. Results and Discussion

The first stage of this research resulted in an analysis of user requirements obtained through interviews, which were subsequently applied to the development of the Teaching Assistant Information System. The interviews were conducted with the Academic and Student Administration Bureau (BAAK) of STT-NF, which is responsible for managing academic administration related to teaching assistants. This requirement analysis aims to identify real user needs and existing problems in the current teaching assistant management process.

3.1. User Story and Iteration Planning

The results of the requirement analysis were then translated into User Stories, which serve as a fundamental component in the Extreme Programming (XP) methodology. A User Story represents a simple textual narrative describing system features from the user's perspective, focusing on what the system should do rather than how it should be implemented [13]. This approach enables effective communication between developers and users and ensures that system development aligns with actual user expectations.

The developed User Stories define the complete functional requirements of the Teaching Assistant Information System. These requirements include teaching assistant registration, lecturer data management, teaching assistant vacancy creation by lecturers based on assigned courses, recruitment management, course and class management, teaching assistant honorarium management, and the development of a system dashboard to support monitoring and decision-making processes. By organizing system requirements into User Stories, the development process becomes more structured and adaptable to changes, which is a key characteristic of the Extreme Programming methodology.

Table 1. User Stories of the Teaching Assistant Information System

ID	Title	Description	Acceptance Criteria
US-01	Login	As a user, I am required to enter system data and can only do so if I have authorized access to log in to the information system.	The system provides a login feature that can only be accessed by users with registered usernames and passwords.
US-02	Manage Lecturer Data	As a user, I can manage lecturer data.	The system provides a form for entering and managing lecturer data.
US-03	Manage Teaching Assistant Data	As a user, I can manage teaching assistant data.	The system provides a form for entering and managing teaching assistant data.
US-04	Manage Course Data	As a user, I can manage course data.	The system provides a form for entering and managing course data.
US-05	Teaching Assistant Recruitment	As a user, I can manage teaching assistant recruitment data.	The system provides a form for entering and managing recruitment data for teaching assistants.
US-06	Class Scheduling	As a user, I can manage class schedule data.	The system provides a form for entering and managing class schedule data.
US-07	Teaching Assistant Honorarium Calculation	As a user, I can manage teaching assistant honorarium data.	The system provides a feature to calculate teaching assistant honorariums.
US-08	Dashboard	As a user, I can view the Teaching Assistant Information System dashboard.	The system provides a dashboard displaying summarized information.
US-09	Logout	As a user, I want to log out of the system after completing my activities in the information system.	The system provides a logout feature that redirects users to the login page.

Table 1 presents the results of the User Stories for the Teaching Assistant Information System to be developed. Each User Story describes specific user roles, expected system functionality, and the objectives to be achieved. The use of User Stories facilitates prioritization of features and supports iterative development, allowing the system to be enhanced incrementally based on user feedback. This approach also helps reduce misunderstandings during development and ensures that implemented features directly address user needs.

After defining the user stories, the next stage is the planning phase of the Teaching Assistant Information System. This phase involves designing system development iterations using the Extreme Programming (XP) methodology. XP is a software development method that emphasizes short development cycles, intensive communication between the development team and users, and continuous feedback [14].

The development of the Teaching Assistant Information System was conducted through three iterations over a total implementation period of six months. The details of each iteration are presented in Table 2.

Table 2. Iteration Details

Iteration	Duration	Task	Subtask	Points	Velocity
1	1 month	Master Data	CRUD Lecturer Table	3	12
			CRUD Teaching Assistant Table	3	
			CRUD Course Table	3	
			CRUD Class Table	3	
2	3 months	Teaching Assistant Recruitment	Teaching Assistant Recruitment Process	9	18
		Teaching Assistant Honorarium	Honorarium Calculation	9	
3	2 months	Dashboard	Information System Dashboard	6	18
		Authentication	User Profile	3	
			Teaching Assistant Registration	6	
			Logout	3	

Notes:
Points indicate task complexity:
1 = Very Low, 3 = Low, 5 = Moderate, 7 = High, 9 = Very High

3.2. Use Case Diagram

The outcome of the subsequent stage is the analysis of the requirements for the teaching assistant information system, which is represented in the form of a Use Case Diagram. A Use Case Diagram is one of the UML diagrams used to illustrate interactions between users and the system [15]. Figure 2 presents the Use Case Diagram of the Teaching Assistant Information System (SIASDOS).

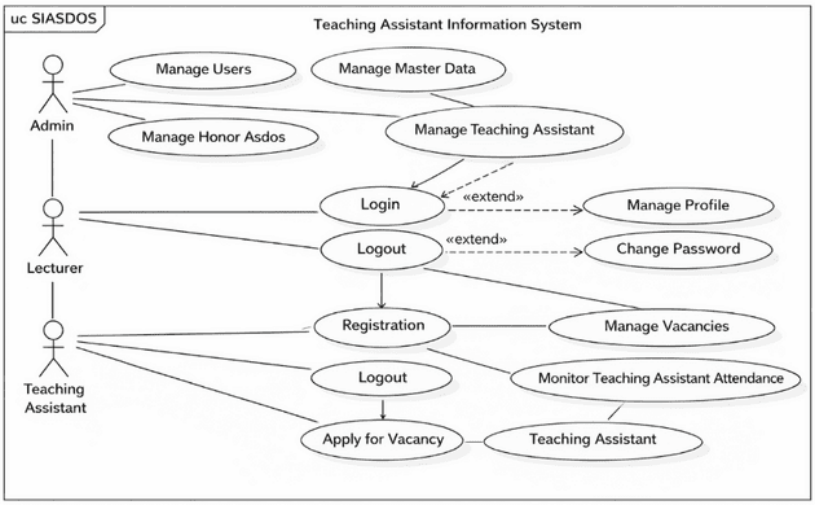

Figure 2. Use Case Diagram of SIASDOS

As shown in Figure 2, there are three actors involved: admin, lecturer, and teaching assistant. To access the Teaching Assistant Information System, all actors must first be authenticated by the system. The admin is responsible for managing user data, lecturers, teaching assistants, courses, and classes. The lecturer can conduct the teaching assistant recruitment process and monitor class activities as well as teaching assistant performance. Meanwhile, the teaching assistant can apply for teaching assistant vacancies opened by lecturers, manage the classes they assist, and view the honorarium they receive.

3.3. ERD Design

The results of the previous stage are then continued to the design phase. At this stage, database design is carried out using an Entity Relationship Diagram (ERD), which is an essential component of database design and serves as a foundation for supporting systematic software development [16]. Figure 3 illustrates the relationships among tables in the Teaching Assistant Information System.

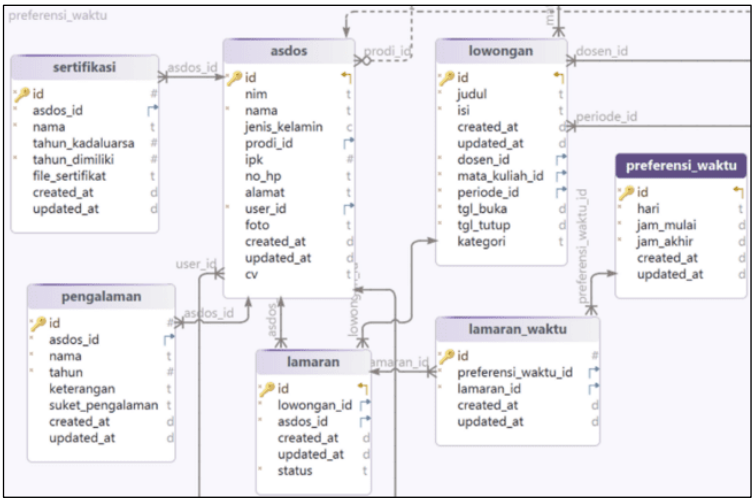

Figure 3. ERD of SIASDOS

The ERD of SIASDOS shown in Figure 3 illustrates the relationships among database tables, where the lecturer table is related to the teaching assistant table in the context of the recruitment process and class scheduling for courses taught by lecturers. In addition, the teaching assistant table is related to the honorarium table, which records the honorarium received by teaching assistants based on the classes they assist.

3.4. Implementation (Model, Controller, View, and User Interface)

The results of the previous stage are then continued to the coding phase. This phase presents several implementation results from the development of the Teaching Assistant Information System using the Laravel Framework by applying the Model, View, Controller (MVC) architectural pattern.

Model

The following code illustrates the implementation of the Teaching Assistant Model, which represents the asdos table along with its attributes and relationships with other related tables in the teaching assistant recruitment process. The model defines the structure of teaching assistant data, including personal information, academic background, and associated user accounts. It also establishes relationships with other entities using Laravel Eloquent ORM to support data integrity and relational mapping. Listing 1 shows the implementation of the Asdos model using the Laravel framework, which represents the teaching assistant entity and its relationship with the user table.

Listing 1. Implementation of the Teaching Assistant Model

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class Asdos extends Model
{
    use HasFactory;
    protected $table = 'asdos';
    protected $fillable = ['nim', 'nama', 'jenis_kelamin', 'prodi_id', 'ipk',
                           'no_hp', 'alamat','user_id', 'foto', 'cv'];
    public function user(): BelongsTo
    {
        return $this->belongsTo(User::class);
    }
 }
```

The Asdos model functions as the Model layer in the MVC architecture, handling data representation and database interaction related to teaching assistants.

View

The following code snippet represents an example of the View implementation, which is responsible for presenting teaching assistant vacancy information to users in a user-friendly interface. Laravel Blade templating is used to separate presentation logic from business logic, ensuring maintainability and readability of the user interface. Listing 2 illustrates the Blade view implementation for the teaching assistant vacancy page, which allows teaching assistants to view and select available vacancies.

Listing 2. Teaching Assistant Vacancy Page View Implementation

```
@extends('index')
@section('content')
    <div class="all-title-box">
        <div class="container text-center">
            <h1>Lowongan Asdos<span class="m_1">
            Pilih Lowongan Asdos yang Akan Kamu Lamar</span>
            </h1>
        </div>
    </div>
@endsection
```

The view functions as the View layer in the MVC architecture, displaying information dynamically based on data provided by the controller.

Controller

The following code shows an example of the Controller implementation, which manages the logic and workflow of teaching assistant activities, particularly in handling teaching assistant vacancies. The controller retrieves open and closed vacancies based on the current date and organizes the data before sending it to the view layer. Listing 3 shows the implementation of the LowonganController, which retrieves open and closed teaching assistant vacancies based on the application period and displays them on the vacancy page.

Listing 3. Teaching Assistant Vacancy Controller Implementation

```
<?php
class LowonganController extends Controller
{
    public function index()
    {
        $lowongan_terbuka = Lowongan::where('tgl_buka', '<=', now())
            ->where('tgl_tutup', '>=', now())
            ->orderBy('id', 'desc')
            ->get();
        $lowongan_tertutup = Lowongan::where('tgl_tutup', '<', now())
            ->orderByDesc('id')
            ->get();
        return view('lowongan', compact('lowongan_terbuka',
                                        'lowongan_tertutup'));
    }
}
```

This controller acts as the Controller layer, bridging the Model and View by processing business logic and controlling data flow within the system.

User Interface

After the coding phase was completed, the next step was designing the user interface (UI) of the system. The Teaching Assistant Information System (SIASDOS) can be accessed through its official website at https://siasdos.nurulfikri.ac.id/. Figure 4 presents the main page of the Teaching Assistant Information System at STT Terpadu Nurul Fikri.

Figure 4. Main Page of SIASDOS

Figure 5 illustrates the class schedule page managed by teaching assistants. Through this page, teaching assistants can record and recap class activities for each meeting.

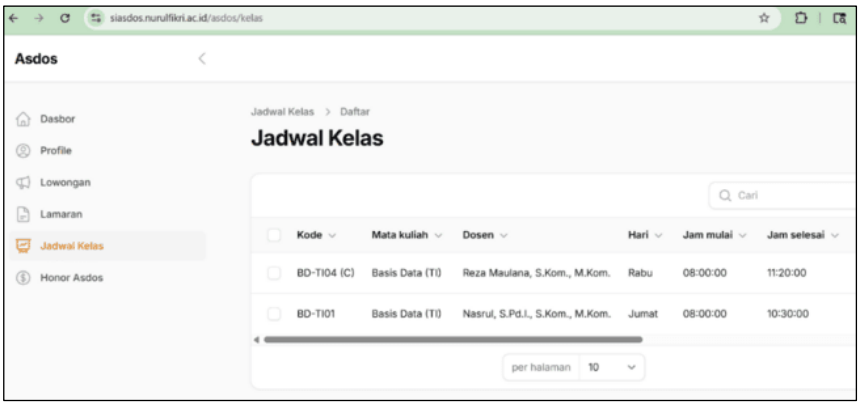

Figure 5. Teaching Assistant Class Schedule Page

Figure 6 displays the honorarium page for teaching assistants. On this page, teaching assistants can filter class schedules based on a specific time range to view the corresponding honorarium received.

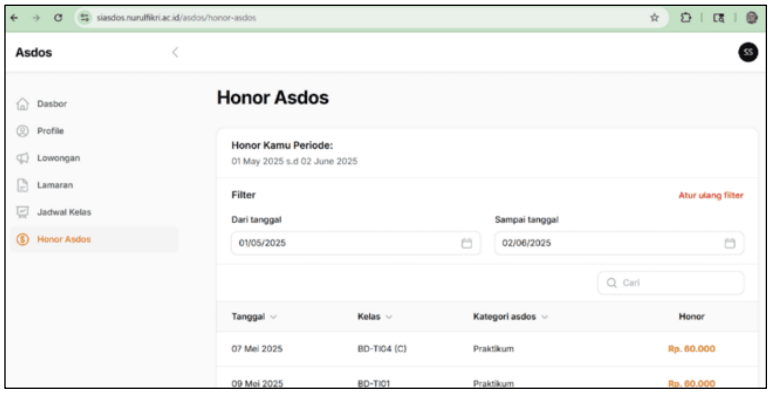

Figure 6. Teaching Assistant Honorarium Page

Compared to the previous manual process, the proposed system reduced data processing time for recruitment and honorarium calculation, minimized data redundancy, and improved scheduling accuracy. User feedback from administrative staff indicated improved ease of monitoring and reduced workload. Although the improvements are based on qualitative evaluation, these findings suggest that the system not only fulfills functional requirements but also enhances operational efficiency.

3.5. Testing and Evaluation (Black Box Testing)

System testing was conducted to ensure that the Teaching Assistant Information System functions according to the defined requirements. The testing method applied in this study is Black Box Testing, which focuses on evaluating system functionality based on input and output behavior without examining internal code structure.

Black Box Testing was selected because it aligns with the user-oriented nature of the system and allows validation of functional requirements from the perspective of end users. The testing process was performed on the main functional modules of the system, including authentication, data management, recruitment, scheduling, and honorarium calculation.

The testing scenarios were derived from the user stories and system requirements identified in the planning phase. Each test case was executed by providing valid and invalid inputs to verify whether the system responses matched the expected outcomes.

Table 3. Black Box Testing Results

No	Tested Feature	Test Scenario	Expected Result	Actual Result	Status
1	Login	User enters valid username and password	User successfully logs into the system	System allows access	Passed
2	Login	User enters invalid credentials	System displays error message	Error message displayed	Passed
3	Teaching Assistant Registration	User completes registration form	Data saved successfully	Data stored in database	Passed
4	Vacancy Application	Teaching assistant applies for a vacancy	Application submitted successfully	Application recorded	Passed
5	Class Schedule Management	Teaching assistant views class schedule	Schedule displayed correctly	Schedule shown	Passed
6	Honorarium Calculation	System calculates honorarium based on class data	Correct honorarium generated	Calculation accurate	Passed
7	Admin Dashboard	Admin views system dashboard	System statistics displayed	Data visualized correctly	Passed
8	Logout	User logs out from the system	User redirected to login page	Redirect successful	Passed

Although Black Box Testing confirms that all system functions operate as intended, this study is limited to functional validation. Usability testing, performance benchmarking, and security evaluation were not conducted in this phase. These aspects are identified as important directions for future research to further assess system robustness and user experience.

Based on the Black Box Testing results, all tested functional modules of the Teaching Assistant Information System operated as expected. The system successfully handled user authentication, teaching assistant registration, vacancy application, class scheduling, honorarium calculation, and administrative monitoring.

The testing results indicate that the implementation of the system using the Laravel Framework with Filament and the Extreme Programming (XP) methodology effectively supports functional correctness and system reliability. Continuous testing conducted in each iteration of XP contributed to early detection of errors and improved system stability.

Overall, the evaluation confirms that the Teaching Assistant Information System is ready for operational use and capable of supporting administrative and academic activities related to teaching assistant management at STT Terpadu Nurul Fikri.

4. Conclusion

The development of a web-based Teaching Assistant Information System at STT Terpadu Nurul Fikri (STT-NF) successfully addresses challenges arising from manual management, including time inefficiency, data inaccuracies, and limited monitoring capabilities. By utilizing the Laravel Framework with Filament as the administrative interface, PostgreSQL as the database management system, and applying the Extreme Programming (XP) methodology, the system effectively supports teaching assistant recruitment, class scheduling, and honorarium calculation in an integrated and structured manner. Black Box Testing results indicate that all core system functionalities operated as expected, achieving a 100% success rate, which confirms that the system meets user requirements and is suitable for operational use.

Beyond functional implementation, this study contributes academically by presenting an applied evaluation of Extreme Programming within a higher education administrative context. The results demonstrate improved efficiency and accuracy in academic administrative processes compared to the previous manual approach. However, this study is limited to functional validation; therefore, future work should include usability testing, performance benchmarking, scalability evaluation, and integration with existing academic systems to support broader adoption in similar institutions.

5. Acknowledgement

The authors would like to express their sincere gratitude to LPPM STT Terpadu Nurul Fikri for funding and supporting this research. Appreciation is also extended to the Academic Administration Bureau (BAAK) and the Laboratory Unit (UPT Lab) of STT Terpadu Nurul Fikri for their assistance and support during the system development and data collection processes.

References

[1] S. Hobert and F. Berens, “Developing a digital tutor as an intermediary between students, teaching assistants, and lecturers,” Educ. Technol. Res. Dev., vol. 72, no. 2, pp. 797–818, Apr. 2024, doi: 10.1007/s11423-023-10293-2.

[2] N. C. Arsita and M. Sholahuddin, “Pengaruh Efikasi Diri dan Iklim Organisasi Terhadap Kinerja Dengan Program Mentoring Sebagai Variabel Mediasi (Studi Pada Asisten Dosen Ums),” SEIKO J. Manag. Bus., vol. 8, no. 1, p. 93, Feb. 2025, doi: 10.37531/sejaman.v8i1.8073.

[3] M. A. Novianto and S. Munir, “Analisis dan Implementasi Restful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi,” J. Inform. Terpadu, vol. 8, no. 1, pp. 47–61, Mar. 2022, doi: 10.54914/jit.v8i1.409.

[4] Misna Asqia, “Designing a Presence Information System for Student Mentoring Activities Using the Laravel Framework,” Indones. J. Comput. Sci., vol. 13, no. 5, Oct. 2024, doi: 10.33022/ijcs.v13i5.4311.

[5] F. Magfira, T. Matulatan, N. F. Fahmitra, F. Irawan, and R. Herikson, “Implementation of Model View Controller Architecture in Designing Outcome-Based Education (OBE) Curriculum Management Information System,” BIO Web Conf., vol. 134, p. 05002, Oct. 2024, doi: 10.1051/bioconf/202413405002.

[6] M. N. Saiholau, “Rancang Bangun Backend Website Pemungutan Suara dengan Menggunakan Framework Express.Js,” J. Inform. dan Tek. Elektro Terap., vol. 12, no. 2, Apr. 2024, doi: 10.23960/jitet.v12i2.4261.

[7] S. G. Tetteh, “Empirical Study of Agile Software Development Methodologies: A Comparative Analysis,” Asian J. Res. Comput. Sci., vol. 17, no. 5, pp. 30–42, Feb. 2024, doi: 10.9734/ajrcos/2024/v17i5436.

[8] A. Lestari, J. Akbar, and H. H. Istyarini, “Perancangan SIMA (Sistem Informasi Monitoring Alat) Pada Stasiun Meteorologi Zainuddin Abdul Madjid Menggunakan Extreme Programming,” Juisik, vol. 3, no. 1, pp. 60–76, 2023.

[9] N. Nasrul, H. Saptono, E. Wibowo, and A. Amalia, “Rancang Bangun Sistem Informasi Manajemen Aset Berbasis Web untuk Menghitung Penyusutan Fiskal,” J. Inform. Terpadu, vol. 10, no. 1, pp. 66–72, Mar. 2024, doi: 10.54914/jit.v10i1.1086.

[10] M. Khatam, F. Fenando, and M. Kadafi, “Sistem Informasi Bimbingan Konseling menggunakan Metode Extreme Programming (Studi Kasus : SMK PGRI Tanjung Raja),” J. Softw. Eng. Ampera, vol.

2, no. 3, pp. 181–195, 2021, doi: 10.51519/journalsea.v2i3.133.

[11] A. Fadly Fahmi, M. Fahrezi, N. M. Fikri, and R. Djutalov, “Sistem Informasi Aplikasi Inventory Stok Barang Berbasis Dekstop Menggunakan Metode Extreme Programming Studi Kasus : Huriah Tembakau,” Teknobis Teknol. Bisnis Dan Pendidik., vol. 1, no. 1, pp. 175–185, 2023, [Online]. Available: https://jurnalmahasiswa.com/index.php/teknobis

[12] N. P. Florensia, R.- -, Y.- Patimah, and Y. N. Chusnul Khotimah, “Analisis Pengujian Sistem Pakar Penyakit Selada Menggunakan Metode Black Box dan White Box Testing,” J. Inform. dan Tek. Elektro Terap., vol. 13, no. 2, Apr. 2025, doi: 10.23960/jitet.v13i2.6394.

[13] F. A. Melati, R. Wibi Pradana, and N. Novi Arisa, “Pengembangan Buku Tamu Digital Pada PT XYZ Menggunakan Metode Personal Extreme Programming,” J. Sistim Inf. dan Teknol., pp. 73–81, Jan. 2024, doi: 10.60083/jsisfotek.v5i4.323.

[14] P. M. Putra, “Implementasi Metode Extreme Programming dalam Pengembangan Sistem Informasi Penggajian di Kampus Politeknik Ganesha Guru,” J. Inform. dan Tek. Elektro Terap., vol. 13, no. 2, Apr. 2025, doi: 10.23960/jitet.v13i2.6283.

[15] L. Setiyani, “Desain Sistem: Use Case Diagram Pendahuluan,” in Prosiding Seminar Nasional: Inovasi & Adopsi Teknologi, 2021, pp. 246–260.

[16] Muhammad Fikri, Bagas Muhammad Husain, Irwan Putra Ndruru, Fikarlin Ndruru, and Fasahati Laiya, “Rancang Bangun Sistem Informasi Persediaan Barang Berbasis Website,” J. Ris. Tek. Komput., vol. 2, no. 1, pp. 01–09, Mar. 2025, doi: 10.69714/sp2ps883.